IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Architectural Intelligence Inspired by Vastu

Landge Omkar, Sakhare Chaitanya, Khokrale Vedanti
Prof Raut.S. P., Kale Dipali, Prof Said.S. K.
JCEI's Jaihind College of Engineering Kuran, Maharashtra, India
Landgeomkar71@gmail.com, Chaitanyasakhare39@gmail.com, khokralevedanti@gmail.com
sumedhameher@gmail.com, dipalikale2010@gmail.com, aidsjcoe@gmail.com

Abstract: *This project explores the integration of Artificial Intelligence (AI) with traditional Vastu principles to create smart, harmonious architectural designs. By leveraging AI techniques such as machine learning, computer vision, and natural language processing, the system optimizes spatial layouts and automates Vastu compliance checks, offering personalized recommendations that enhance user comfort, energy balance, and cultural relevance. The approach bridges modern technology with ancient architectural wisdom, enabling efficient, aesthetically pleasing, and spiritually aligned living spaces tailored to individual preferences. This fusion of AI and Vastu is poised to revolutionize interior design by making it more intelligent, adaptive, and culturally sensitive. The platform offers personalized recommendations by learning user preferences and adapts designs for energy efficiency and environmental sustainability. The 30 tactics we identified are aimed to serve as an initial reference guide for further exploration into Green AI from a software engineering perspec- tive, and assist in designing sustainable ML-enabled systems. The platform offers personalized recommendations by learning user preferences andadapts designs for energy efficiency and environmental sustainability.*

CCS CONCEPTS
• *Software and its engineering → Designing software; Soft- ware architectures;*
• *Social and professional topics → Sus- tainability;*
• *Computing methodologies → Machine learning.*

Keywords*:* Software architecture, architectural tactics, ML-enabled systems, environmental sustainability, Green AI.

Lay Abstract: *Machine learning (ML) is a technology field that wants to provide software with functionality similar to human- like intelligence, e.g., for understanding text or describing images. However, creating and using systems with ML needs a lot more computing power than non-ML systems, which is bad for the envi- ronment. Companies therefore need concrete advice on how they can create ML systems that are environmentally sustainable. In this paper, we provide a catalog of 30 green architectural tactics for these systems. An architectural tactic is a high-level design technique to improve software quality, in our case environmental sustainability. To achieve this, we analyzed 51 scientific papers and later discussed with 3 experts to improve and extend our catalog. If many companies start using these tactics, the energy footprint of systems with ML can be greatly reduced.*

I. INTRODUCTION

This project focuses on combining Artificial Intelligence with Vastu Shastra, an ancient Indian system of architecture that emphasizes harmonious spatial design to promote well-being. Using AI techniques like machine learning and computer vision, the system will automate the process of checking Vastu compliance in architectural plans, saving time and reducing errors compared to manual methods. It will intelligently suggest optimized layouts that improve energy flow, comfort, and aesthetic appeal based on individual user preferences and sustainability goals.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

The integration of virtual reality will allow users to virtually walk through and interact with proposed designs, making the design process more immersive and collaborative. This approach modernizes traditional Vastu wisdom by enhancing it with data-driven decision-making and personalized design adaptation, ultimately helping architects, interior designers, and homeowners create smart, sustainable living spaces that respect cultural values.

Additionally, the project will incorporate a user-friendly interface that accommodates both professionals and laypersons, making Vastu-compliant design accessible to a broader audience. By learning from user feedback and evolving design trends, the system will continuously improve its recommendations, ensuring relevance and adaptability over time. This dynamic, AI-driven platform not only enhances architectural creativity and efficiency but also supports cultural preservation by keeping traditional principles alive in contemporary built environments.

Holistically evaluating the impact of an ML-enabled system on en- vironmental sustainability is challenging because the system goes through several stages during its life cycle, from ML model develop- ment to deployment, maintenance, and evolution of the ML-enabled system. However, the negative environmental impact of ML is usu- ally directly related to the required high computational power, i.e., the amount of energy required to train and run ML models [25]. Computational power is typically evaluated via energy consump- tion measured in joules (J) or kilowatt-hours (kWh), computing power via floating-point operations per second (FLOPS), GPU/CPU utilization as a percentage, GPU/CPU hours, response time, or car- bon emissions [20]. Additional factors beyond software-related computational power like hardware manufacturing, transportation, or e-waste are harder to quantify and outside the study scope.

ML models present unique challenges for the development of ML-enabled systems due to their data-dependent nature, dynamic behavior, and lack of transparency [33]. These unique character- istics of ML have affected the fields of software engineering and software architecture, leading to an emerging focus on software architecture for ML-enabled systems or SA4ML. SA4ML aims to develop principles, practices, and tools for improving the develop- ment of ML-enabled systems [39]. Papers on patterns and tactics for ML-enabled systems have started to appear [19, 62], but they concentrate primarily on developing effective ML-enabled systems while lacking the environmental perspective.

Despite the increasing interest in the environmental impact of ML [56], it is still challenging to obtain a comprehensive overview of the recommended best practices for developing green ML-enabled systems. To gain a deeper understanding of the effectiveness and usefulness of different practices, we explore the intersection of Green AI and software architecture by identifying green architec- tural tactics for ML-enabled systems. We achieve this through a literature-based synthesis and a subsequent focus group review with software architecture experts working on SA4ML to validate and extend the initial collection of tactics. In our study, we address the following research questions:

• RQ1: Which green architectural tactics for ML-enabled sys- tems can we synthesize from scientific literature?

• RQ2: How do SA4ML experts perceive our synthesized col- lection of tactics?

Our contribution is a collection of 30 green architectural tactics for ML-enabled systems. To structure the collection, we organize the tactics into six categories that map to different aspects of ML- enabled systems development. Given that a tactic is a design tech- nique used to improve a specific quality attribute [7], we expect our contribution to guide practitioners to make more environmentally sustainable decisions when engineering ML-enabled systems.

II. BACKGROUND AND RELATED WORK

We first provide an overview of important concepts such as ML- enabled systems, environmental sustainability, and architectural tactics, and then discuss related work in the area.

2.1 ML-Enabled Systems

ML is a sub-field of AI that concentrates on developing algorithms capable of learning from data [46]. ML models are trained on in- put data to produce a desired outcome without being explicitly programmed to do so: they automatically identify connections in the data and learn complex patterns [14]. These learned patterns can then be used to make predictions or decisions for new, previ- ously unseen data. According to Amershi et al. [3], building an ML model usually follows a specific development process with feedback loops: understanding business goals, preparing the data, designing the model, training the model, testing the model, deploying the model in the production environment, and finally managing and monitoring it during usage.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

While ML models can be used for stand-alone data science or analytics activities, they are also frequently embedded into larger software systems. The term ML-enabled system therefore refers to a software system that includes at least one ML component [33]. Due to the special characteristics of machine learning, integrating ML components into software systems comes with additional chal- lenges in terms of software architecture and design, e.g., strong dependencies on training data, lack of transparency for ML com- ponent behavior, and abrupt changes in the prediction quality of ML components [33]. Ensuring quality is therefore an important and complicated activity for ML-enabled systems, with many qual- ity attributes (QAs) to consider [17]. The unique characteristics of ML model and system development have also given rise to special system-level QAs, which include, e.g., prediction accuracy, explain- ability, monitorability, sustainability, prediction cost, training cost, and training latency [21, 43, 49, 57].

2.2 Environmental Sustainability of ML

Holistically evaluating the impact of an ML-enabled system on en- vironmental sustainability is challenging because the system goes through several stages during its life cycle, from ML model develop- ment to deployment, maintenance, and evolution of the ML-enabled system. However, the negative environmental impact of ML is usu- ally directly related to the required high computational power, i.e., the amount of energy required to train and run ML models [25]. Computational power is typically evaluated via energy consump- tion measured in joules (J) or kilowatt-hours (kWh), computing power via floating-point operations per second (FLOPS), GPU/CPU utilization as a percentage, GPU/CPU hours, response time, or car- bon emissions [20]. Additional factors beyond software-related computational power like hardware manufacturing, transportation, or e-waste are harder to quantify and outside the study scope.

The development stage of ML models is usually regarded as the most energy-intensive phase within the life cycle of ML-enabled systems [25], especially for systems using generative AI, such as in the form of large language models (LLMs) [24], since these models require large amounts of data. However, increasing the number of data points is also used as a general strategy to train more accurate ML models, which makes data preprocessing and model training consume more energy [63]. Training data may also be complex and multidimensional, which is why data cleaning, labeling, and feature engineering can be energy-intensive [64]. Along with the increased amount of data, ML model size and complexity also in- crease. As a result, the model training and tuning phases are very energy-intensive, especially if multiple rounds of testing are re- quired [47]. For example, many deep learning models are often trained for long periods of time, ranging from dozens to thousands of hours, sometimes using powerful but energy-intensive hardware such as GPUs or other processing units optimized for ML work- loads [8]. While ML training receives the majority of attention, each development stage in the ML development process has its own concerns and opportunities related to energy consumption and en- vironmental sustainability. Therefore, evaluating the sustainability of ML requires a holistic view of the ML life cycle [63].

2.3 Green Architectural Tactics

Synthesizing generalizable design decisions to achieve certain QAs is an important practice in the software architecture community to enable the reuse of architecture knowledge.

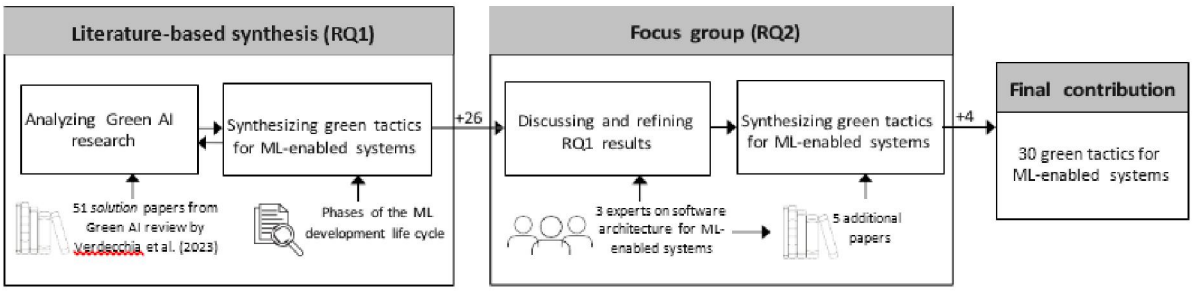

Figure 1: Study Design and Execution

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

One concept in this space is an architectural tactic (or tactic in short form), i.e., a high-level design decision focusing on achieving a single QA for a software system [7]. Tactics differ from design patterns [15]. Patterns typically are more concrete and cover multiple decisions, tactics, and QAs, often through trade-offs that impact some QAs positively and others negatively. In this sense, tactics are high-level building blocks from which patterns are composed, but choosing an architecture pattern can also guide the selection of complementary architectural tactics [18]. Harrison and Avgeriou [18] discuss the relationship between tactics and patterns in greater detail.

Architectural tactics often improve one primary QA and are typically organized in QA-related collections such as security or performance tactics [40]. However, tactics may also have a positive influence on other secondary QAs, even though these QAs may not be the main focus of the tactic. For example, a tactic aimed at improving performance via less resource-intensive computation and faster response times will often also improve energy efficiency. In the context of ML-enabled systems, we focus on green tactics, i.e., tactics that have a positive impact on environmental sustainability, usually in the form of improving energy or carbon efficiency. All architectural tactics are design decisions that positively impact a QA of a system or part of a system, and can manifest themselves within different scopes. For example, we later introduce the tactics Remove redundant data (T2) and Decrease model complexity (T9). These tactics do not focus on the complete software system, but rather focus solely on the data (T2) and model (T9) as their scope. This interpretation is in line with the definition of software architecture as the set of significant design decisions [22].

2.4 Related Work

Architectural tactics are a popular medium to convey design knowl- edge in the software engineering research community. A recent mapping study by Márquez et al. [40] identified a total of 91 pri- mary studies. However, the authors criticized that most studies use the concept of tactics in a way that is not in line with the original definition, and that it is often unclear from which data sources the tactics were synthesized. They also state that they found little evidence of the use of architectural tactics in industry.

While sustainability or energy efficiency are not explicitly cov- ered by the primary studies discussed by Márquez et al. [40], some authors have proposed green tactics for different domains. Procac- cianti et al. [44] and Vos et al. [58] synthesized tactics for optimizing the energy efficiency of software running in the (public) cloud. Simi- larly, Chinnappan et al. [11] and Malavolta et al. [34] conceptualized tactics for energy-aware robotics software. While these could be situationally useful for ML-enabled systems in such domains, it is important to have tactics that take the specifics of ML into account. By now, many publications exist on the general topic of Green AI. A recent review by Verdecchia et al. [56] collected 98 studies in this fast-growing field. However, the majority of these publications concentrate on understanding the mechanisms related to the energy efficiency of ML and do not provide their findings in an easily accessible and actionable form. Moreover, the existing techniques are distributed among many different publications, making their holistic consideration difficult. In this sense, a collection of concrete tactics that practitioners can use to make ML-enabled systems more sustainable does not currently exist. One work that starts to produce actionable guidance is a study by Shanbhag et al. [48]: they used the concept of design patterns to synthesize eight "energy patterns" for deep learning projects from an analysis of Stack Overflow posts. The patterns were subsequently validated with a questionnaire survey with 14 practitioners. While their catalog is a promising step in the right direction, our study aimed for a broader scope and coverage: we included all forms of ML, derived our tactics from scientific literature, and took a holistic look at the life cycle of ML-enabled systems. This ultimately led to the 30 green tactics we synthesized, which also include the 8 energy patterns from Shanbhag et al. [48].

III. STUDY DESIGN AND EXECUTION

We designed our study with a qualitative research approach orga- nized in two steps (see Fig. 1). In the first step, we used scientific literature on Green AI to synthesize green architectural tactics for ML-enabled systems (RQ1). To structure the tactics, we formed categories and iteratively refined the collection. Once the collection reached sufficient maturity, we carried out the second step, in which we conducted a focus group with three SA4ML experts (RQ2). This allowed us to validate and further refine the collection of tactics.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

A focus group is a qualitative method to collect opinion-based data about a specific topic through a systematic discussion with participants knowledgeable about the topic [28]. The number of participants needs to be small enough for everyone to contribute, and a researcher has to carefully moderate the exchange, typically by following a predefined protocol with questions. Based on such a group setting, exchanges between participants can stimulate new ideas and lead to stronger consensus. Using the focus group results, the collection was substantially improved and extended.

3.1 Literature-Based Synthesis (RQ1)

The data collection for the first part of the study was based on a recent systematic literature review on Green AI by Verdecchia et al. [56]. They queried the publisher-agnostic search engines Google Scholar, Scopus, and Web of Science, and complemented this with an iterative snowballing process, leading to a total of 98 primary studies. These studies were categorized into observational, solu- tion, and position articles. The solution category included papers providing techniques or tools to address Green AI issues, e.g., to improve the energy efficiency of ML models or components. Be- cause these papers represent promising sources for architectural tactics, we used all 51 papers from this category as the start for our synthesis. Papers from the other two categories (observational and position) did not contain relevant information to synthesize actionable techniques.

Following the guidelines of Bowen [9], all 51 selected papers were evaluated with qualitative data analysis methods, mainly document analysis and content analysis. As a first step, the main researcher performed a document analysis for each paper that included skim- ming, reading, and interpretation, which was followed by a content analysis to identify and synthesize green tactics for ML-enabled systems. This content analysis was guided by the "Awesome Tac- tic" template [32] from the Archive of Awesome and Dark Tactics (AADT). AADT is a curated, open-source knowledge base that pro- vides information on tactics for various domains that can have either a positive or negative impact on software quality. The AADT template for tactics is grounded in foundational literature about architectural tactics and includes fields such as tactic intent, partic- ipant, context, primary QA, secondary QAs, and measured impact. Using descriptive coding methods [36], the synthesized tactics were organized into categories representing different aspects of the devel- opment life cycle of ML-enabled systems. The analysis took place in an iterative manner, with frequent reviews and exchanges between researchers in the spirit of continuous refinement.

3.2 Focus Group (RQ2)

In the second part of the study, we conducted a focus group with three experts in software architecture for ML-enabled systems (SA4ML). As is common with focus groups [28], we used purposive sampling to recruit participants from our network whom we knew to be very knowledgeable and experienced about the topic. All three experts were in senior, research-related roles with extensive expertise in software architecture, ML-enabled systems, and soft- ware sustainability. One participant worked at a university and two participants at a research institute. The experts did not have ML engineering roles in industry, but all of them had extensive knowl- edge of the state of ML engineering in industry through frequent collaborations and projects with companies or public organizations that developed or acquired ML-enabled systems. Two authors were present during the focus group as moderators. The session took place as a video call of roughly 75 minutes, which was recorded for later analysis with everyone's permission.

Following a protocol with questions, the moderators presented the collection of synthesized green tactics and encouraged an open discussion to receive feedback. One aim was to validate the tac- tics' soundness, relevance, and applicability to industry scenarios. Additionally, the protocol included open questions to find out if important tactics applied in industry were missing. This led to an in-depth discussion about the tactics and their categorization. The participants suggested several additional publications to extend the collection. These papers were included in the research to extract possible tactics from them.

Afterward, the recording of the focus group was analyzed to identify the key themes of the discussion. The main topics were written down and used to improve the tactics identified in the first step of the study. All the feedback received about the identified tactics was considered, and, if applicable, the tactics were modified accordingly. The additional papers suggested by the participants were analyzed using the same document and content analysis ap- proach described

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

in Section 3.1 to synthesize additional green tactics for ML-enabled systems. In particular, the focus group added the following contributions:

Extension of the catalog: as mentioned, the experts suggested additional papers that led to the identification of four new tactics.

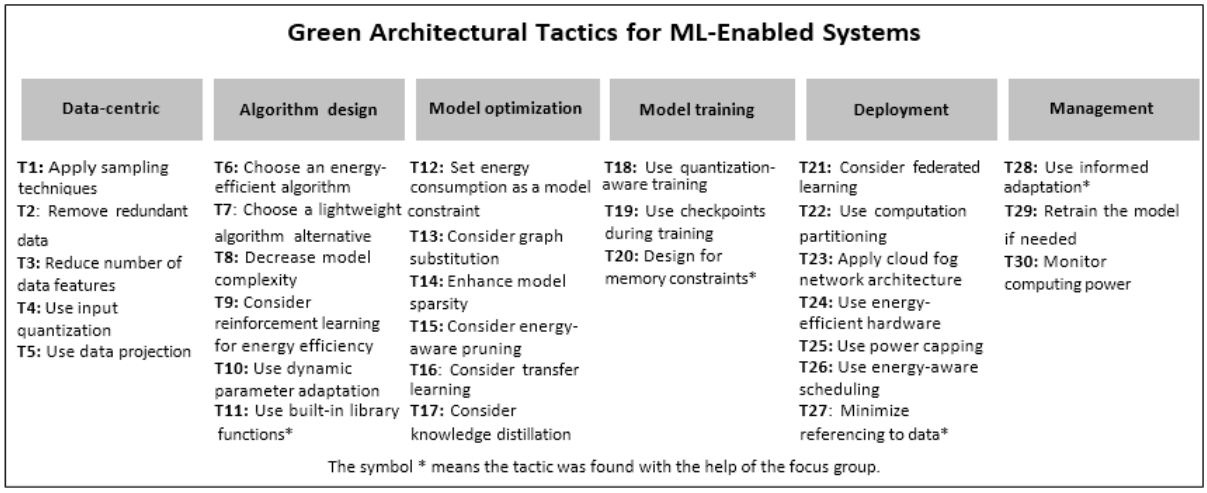

Figure 2: Catalog of the 30 Synthesized Green Architectural Tactics for ML-Enabled Systems

Refinements of descriptions and categories: the descriptions and categorization of the tactics benefited from the insights pro- vided by the experts, which observed that: (i) while all tactics are architecture-relevant, many are more model-focused, i.e., related to the ML model rather than the system as a whole. Overall, the tactics are quite diverse and focus on data management, process aspects, or design vs. implementation aspects to various degrees; (ii) most tactics target energy efficiency as a primary QA, but some also focus on other QAs or include important trade-offs with secondary QAs; (iii) some tactics are specific to certain ML techniques and, as such, are not universally applicable to achieve energy efficiency. This led to renaming some tactics from "Use . . . " to "Consider . . . ". For example, Consider transfer learning (T16) can reduce energy consumption, but it is not always applicable; so one might say that, if applicable, T16 is preferred to increase energy efficiency.

Observations and reflections for future work: the experts provided interesting observations and reflections that point to pos- sible follow-up work. These are discussed in Section 5.

3.3 Threats to Validity

According to Verdecchia et al. [55], threats to validity are often considered as an afterthought in the SE community, and trade-offs among threats to validity are often neglected. For this study, we prioritized the soundness and relevance of the collection. There- fore, we consciously accepted limitations in our study design that decreased external validity, e.g., completeness and generalizability, but improved internal validity, e.g., soundness and consistency.

In this regard, one threat to validity is that our data collection relied on the literature review of Verdecchia et al. [56]. While the authors followed a sound protocol with extensive snowballing, their initial search terms were focused on Green AI in general, and did not include terms related to ML-enabled systems or software ar- chitecture. This could have led to some papers relevant to green architectural tactics for ML-enabled systems not being found. How- ever, the subsequent focus group partly mitigated this threat by suggesting additional literature.

The focus group method can also be prone to some threats to validity. Because it involves relatively few participants, the general- izability of the results can be impacted. However, this also allows an in-depth discussion of the topic with rich, qualitative reflections. Additionally, we ensured that the participants were prestigious experts on the topic, with extensive experience and expertise.

Lastly, qualitative research can be prone to subjective biases that can influence the results. To mitigate this, the collection of tactics was not only validated by the focus group, but also extensively reviewed and refined by the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

research team until consensus was reached. During these iterative refinements, great care was taken to preserve clarity and consistency, e.g., through proofreading of a refined tactic by someone else. Three researchers were involved in the in-depth synthesis and refinement of the tactics, with the rest providing high-level feedback. All in all, we do not claim com- pleteness for our final collection of tactics, but we demonstrate a representative list of actionable and relevant tactics that can sup- port practitioners in creating more environmentally sustainable ML-enabled systems.

IV. RESULTS

The initial literature-based synthesis before the focus group dis- cussion resulted in 26 tactics. Based on the focus group, we val- idated these tactics and identified four additional tactics via the literature suggested by the experts (T11, T20, T27, and T28). Our final collection therefore consists of 30 green architectural tactics for ML-enabled systems. They are organized into six categories: data-centric, algorithm design, model optimization, model training, deployment, and management. These categories correspond to the different types of artifacts and phases relevant to the life cycle of ML-enabled systems so that tactics can be selected based on current status or for certain parts of the system. A summary of the resulting tactics is shown in Fig. 2. The collection is also available in the on- line repository that accompanies this paper1 and in the Archive of Awesome and Dark Tactics.2 To illustrate and explain the used tac- tic template, we present one example in full detail below (see Fig. 3 for the visual template). The remaining tactics will be presented in short summaries in their respective category subsections.

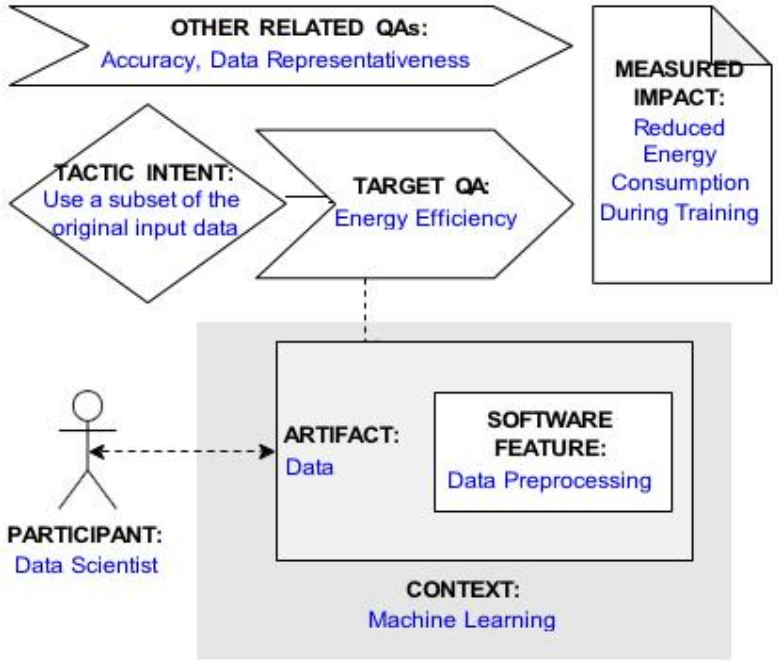

Figure 3: Tactic Template for Apply Sampling Techniques (T1)

The tactic Apply sampling techniques (T1) primarily intends to improve energy efficiency (Target QA), but also impacts accuracy and data representativeness (Other Related QAs). In the template, the impact for the Target QA is always positive, but for Other Related QAs, it can also be negative, e.g., T1 can reduce accuracy. The Tactic Intent summarizes the general technique that is applied, i.e., “use a subset of the original input data” for T1. In the Context of “machine learning”, the targeted Artifact of T1 is “data”. Other tactics target, e.g., models, training algorithms, or deployment infrastructure in- stead. The related Software Feature of T1 is “data preprocessing”, with other tactics focusing on, e.g., model training or inference. T1 is usually applied by the Participant “data scientist”. For tactics with empirical evidence, the Measured Impact is reported, i.e., “reduced energy consumption during training” for T1. The textual template also includes the references to the respective studies.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

4.1 Data-Centric

This category includes tactics that process input data for ML models to promote energy efficiency, shown as tactics T1-T5 in Table 1.

Table 1: Data-Centric Green Tactics for ML-Enabled Systems

Tactic	Description	Target QA	Source
T1: Apply sampling techniques	Use a smaller subset of the original input dataset	Energy efficiency	[54][61]
T2: Remove redundant data	Detect and remove redundant data from the original input data	Energy efficiency	[5][13]
T3: Reduce number of data features	Reduce the number of input data features used	Energy efficiency	[54]
T4: Use input quantization	Convert input data to smaller precision	Accuracy*	[1][26]
T5: Use data projection	Project data into a lower-dimensional embedding	Performance*	[45]

The * means energy efficiency was considered a secondary QA

T1: Apply sampling techniques. The size of the input data has a positive correlation with the energy consumption of com- puting [4]. Therefore, reducing the size of the input data can have a positive impact on the energy efficiency of ML. Sampling tech- niques, such as simple random sampling or systematic sampling, offer different approaches to selecting a subset of the input data. Verdecchia et al. [54] employed stratified sampling to reduce the number of data points. Stratified sampling means randomly select- ing data points from homogeneous subgroups of the original dataset. This technique resulted in savings in energy consumption [54].

T2: Remove redundant data. This tactic aims to decrease the size of the input data, which consequently reduces the size of the model. Identifying and removing redundant data for ML models can decrease computing time, the number of computations, energy consumption, and memory space. Redundant data refers to those data points that do not contribute significantly to the accuracy of the model. Therefore, removing these unimportant data points does not sacrifice much accuracy. Removing redundant data reduces the energy consumption of training and inference [13, 52].

T3: Reduce number of data features. A large number of data features in the ML model can lead to high computing power re- quirements during training and inference. Typically, ML scenarios involve a huge number of features or variables that describe the input data. However, not all of these features are necessary for the model to make accurate predictions. Therefore, reducing the number of input data features can lead to improved energy effi- ciency while maintaining accuracy. This reduction can be achieved by selecting only a subset of all the available data features [54].

T4: Use input quantization. Input quantization in ML refers to converting the data to a smaller precision, e.g., reducing the number of bits used to represent the data. According to a study by Abreu et al. [1], using 10-bit precision is sufficient for achieving accuracy in ML models, and using more does not contribute to accuracy.

Therefore, using higher precision in this case is a waste of resources. Data quantization can also be used in federated learning to reduce energy consumption and memory access [26]. Using precise data values through input quantization can even have a positive impact on the accuracy of the ML model by reducing overfitting.

T5: Use data projection: Data projection means transforming data into a lower-dimensional embedding. Reducing the dimensionality of input data shrinks the dimensionality of deep neural networks (DNN), which leads to improved

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

performance of the model. Using data projection as a preprocessing step can result in energy improvements without sacrificing performance or accuracy [45].

4.2 Algorithm Design

Tactics in the algorithm design category refer to decisions that are made when designing the ML model. This category consists of six different tactics (T6-T11) that are shown in Table 2.

Table 2: Green Tactics Related to Algorithm Design

Tactic	Description	Target QA	Source
T6: Choose an energy-efficient algorithm	Choose the most energy-efficient algorithm that achieves sufficient level of accuracy	Energy efficiency	[25]
T7: Choose a lightweight algorithm alternative	If possible, choose lighter alternatives of existing algorithms	Energy efficiency	[50]
T8: Decrease model complexity	Decrease the complexity of an ML model	Energy efficiency	[2][38]
T9: Consider reinforcement learning for energy efficiency	Use reinforcement learning to optimize energy efficiency at run time	Energy efficiency	[27][37]
T10: Use dynamic parameter adaptation	Design parameters that are dynamically adapted based on the input data	Energy efficiency	[16]
T11: Use built-in library functions	Use built-in libraries for ML models if possible	Performance*	[48]

The * means energy efficiency was considered a secondary QA

T6: Choose an energy-efficient algorithm. Different ML al- gorithms have different levels of energy consumption and com- putational power. For example, K-nearest neighbor (KNN) algo- rithms have a much higher energy consumption than Random Forest (RF) [54]. High energy consumption does not necessarily mean that the algorithms perform better or achieve higher accuracy levels than low-energy algorithms. Thus, choosing suitable, energy- efficient algorithms that achieve wanted outcomes can reduce the energy consumption of ML models [25].

T7: Choose a lightweight algorithm alternative. Some al- gorithms may have lightweight alternatives. Using these lighter models can have a lower energy consumption without sacrificing other important QAs. For example, spiking neural networks (SNN) are seen as a lightweight and energy-efficient alternative to convo- lutional neural networks (CNN). CNN models can be converted to SNN without a significant loss of accuracy or performance [50].

T8: Decrease model complexity. Complex ML models have been shown to have high energy consumption, and therefore scal- ing down the model complexity can contribute to environmental sustainability. Simplifying the model structure can lead to faster training and inference times, making it more efficient to deploy and use in real-world

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

applications. For example, using a simple three-layered Convolutional Neural Network architecture [38] and shallower Decision Trees [2] has shown to be energy-efficient while still providing high levels of precision.

T9: Consider reinforcement learning for energy efficiency. Algorithms can be designed to optimize energy efficiency through reinforcement learning. Reinforcement learning receives feedback on its actions and adjusts its behavior accordingly. Reinforcement learning models can be used to identify the most energy-efficient options in real time and make informed decisions based on that information [27, 37]. Additionally, other QAs can also be targeted for optimization, e.g., accuracy or CPU/RAM usage.

T10: Use dynamic parameter adaptation. Dynamic parameter adaptation means that the hyperparameters of an ML model are dynamically adapted based on the input data, instead of determining the exact parameters values in the algorithm. For example, García- Martín et al. [16] used an nmin adaptation method for very fast decision trees. The nmin method allows the algorithm to grow faster in those branches where there is more confidence in creating a split, and delaying the split on the less confident branches. This method resulted in decreased energy consumption.

T11: Use built-in library functions. Apply built-in library functions in the ML model instead of writing custom implementa- tions. The existing built-in library functions are usually optimized and well-tested, which is why they may have improved perfor- mance and energy efficiency compared to custom-made functions. For example, these built-in libraries can be used for tensor opera- tions [48].

4.3 Model Optimization

The model optimization category includes all the tactics that are re- lated to the model optimization stage of the ML model development process. These tactics (T12-T17) are shown in Table 3.

T12: Set energy consumption as a model constraint. This tac- tic sets a predetermined energy consumption threshold for the ML model optimization process. The optimization takes into account the energy consumption of the model during both the optimization and training phases. The objective is to train the model in a way that it stays within the specified energy consumption threshold. This approach views model optimization as an optimization prob- lem, where for instance hyperparameters and the model itself are optimized based on predetermined limits [59, 66].

T13: Consider graph substitution. In the context of deep neural networks (DNN), substitution refers to replacing a large model with a smaller one that performs a similar task. Energy- aware substitution, however, means replacing energy-intensive nodes of DNNs with less energy-consuming nodes. For example,

Table 3: Green Tactics Related to Model Optimization

Tactic	Description	Target QA	Source
T12: Set energy consumption as a model constraint	Consider energy consumption as one predetermined parameter for optimizing the ML model	Energy efficiency	[59][66]
T13: Consider graph substitution	Replace energy-intensive model parts with similar, but less energy-consuming parts	Energy efficiency	[60]
T14: Enhance model sparsity	Reduce the number of model parameters or set their values to zero	Energy efficiency	[68]
T15: Consider energy-aware pruning	Prune neural networks starting from the most energy-intensive layer	Energy efficiency	[67]
T16: Consider transfer learning	Use pre-trained ML models for other similar tasks	Energy efficiency	[23][48]
T17: Consider knowledge distillation	Use knowledge from a large ML model to train a smaller model	Performance*	[48][66]

The * means energy efficiency was considered a secondary QA

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

Wang et al. [60] showed energy savings of 24% in their study of energy-aware graph substitution.

T14: Enhance model sparsity. Enhancing the sparsity of an ML model means reducing the number of model parameters or setting their values to zero. For example, weight sparsification involves identifying and removing unnecessary or less important weights in a neural network. Enhancing model sparsity decreases the complexity of the model and consequently reduces requirements for storage and memory. Therefore, it also results in lower power consumption [69].

T15: Consider energy-aware pruning. Pruning is the process of reducing the complexity and size of an ML model by removing unnecessary or less important components, such as weight. In energy-aware pruning, energy consumption of a neural network is used to guide the pruning process to optimize for the best energy efficiency. With the estimated energy for each layer in a CNN model, the algorithm performs layer-by-layer pruning, starting from the layers with the highest energy consumption to the layers with the lowest energy consumption. For pruning each layer, it removes the weights that have the smallest joint impact on the output feature maps [66].

T16: Consider transfer learning. Transfer learning means using knowledge gained from a task (pre-trained model) and trans- ferring it to another similar task. This is feasible only if there is an existing pre-trained model available for use. The absence of or reduction in the effort for model training results in savings in energy consumption [23, 48].

T17: Consider knowledge distillation. Knowledge distillation is a technique where a large, complex model (teacher) is used to train a smaller, simpler model (student). The goal is to transfer the learned information from the teacher model to the student model, allowing the student model to achieve comparable performance while requiring fewer computational resources [48, 66].

4.4 Model Training

Three tactics (T18-T20) are related to ML model training, and are shown in Table 4.

Table 4: Green Tactics Related to Model Training

Tactic	Description	Target QA	Source
T18: Use quantization-aware training	Convert high-precision data types to lower precision during training	Accuracy*	[26, 50]
T19: Use checkpoints during training	Use checkpoints to avoid a knowledge loss in case of a premature termination	Recoverability*	[48]
T20: Design for memory constraints	Consider possible memory constraints during training	Recoverability*	[48]

The * means energy efficiency was considered as a secondary QA

T18: Use quantization-aware training. Quantization-aware training is a technique used to train neural networks to convert data types to lower-precision ones. The idea is to use fixed-point or integer representations instead of the more commonly used higher precision floating-point representations. This improves the performance and energy efficiency of the model in federated learn- ing [26, 50].

T19: Use checkpoints during training. Training is an energy- intensive stage of the ML model life cycle, which may take long periods of time. Sometimes, a failure or a hardware error can ter- minate the training process before it is completed. In those cases, the training process has to be started from the beginning and all progress is lost. The use of checkpoints, however, can save progress in regular intervals and in case of a premature termination, the training process can continue from the last checkpoint [48].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

T20: Design for memory constraints. Model training requires memory, and sometimes memory leaks and OOM (out of memory) errors may occur during that process. If that happens, the knowl- edge gained during the prior training process is lost. By considering memory availability constraints and addressing possible OOM ex- ceptions, the model training pipeline can be designed to operate within the available memory limits. It reduces the likelihood of errors and prevents unnecessary energy consumption [48].

4.5 Deployment

The deployment category contains tactics related to deploying an ML-enabled system such that it is more environmentally sustainable. These tactics (T21-T27) are shown in Table 5.

T21: Consider federated learning. Federated learning is an ML approach that aims to train a shared ML model on decentralized devices. Instead of sending raw data to a central server, federated learning trains the model directly on the devices where the data is generated, such as mobile phones or edge devices. Only the trained

Table 5: Green Tactics Related to Model Deployment

Tactic	Description	Target QA	Source
T21: Consider federated learning	Train the model and store data in decentralized devices	Energy efficiency	[26]
T22 Use computation partitioning	Divide computations between a client and a cloud server	Energy efficiency	[35]
T23: Apply cloud fog network architecture	Use an architecture in which the models are processed between end devices and cloud	Energy efficiency	[71]
T24: Use energy-efficient hardware	Use energy-efficient, ML-suitable hardware	Energy efficiency	[25]
T25: Use power capping	Set energy consumption limits for hardware	Energy efficiency	[29]
T26: Use energy-aware scheduling	Dynamically optimize the scheduling of ML tasks	Resource utilization*	[52]
T27: Minimize referencing to data	Avoid unnecessary read and write data operations	Energy efficiency	[48]

The * means energy efficiency was considered a secondary QA

model or updated model parameters are sent to a central server [26]. Federated learning decreases the resources needed for transferring large amounts of data to a central server, which results in improved energy efficiency.

T22: Use computation partitioning. Computation partition- ing is the process of dividing the computations of a CNN between a mobile client and a cloud server. The goal is to optimize energy consumption and efficiency. The NeuPart framework [35] is an example of a partitioning approach. NeuPart divides computational tasks between the mobile device (client) and the remote server or data center (cloud) in real time based on energy consumption. By offioading computationally intensive tasks to the cloud and exe- cuting lighter tasks locally, NeuPart resulted in significant energy savings of up to 52% in cloud-based computations [35].

T23: Apply cloud fog network architecture. Edge devices are usually connected to distant cloud services. However, bringing the cloud closer to edge devices could be more energy efficient. A cloud fog network (CFN) is one way to achieve that [71]. CFN supports an architecture where deep neural network models are processed in servers between

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

end-devices and the cloud. For ex- ample, Yosuf et al. [71] present a CFN architecture that consists of four layers: IoT end devices, Access Fog (AF), Metro Fog (MF) and Cloud Datacenter (CDC). This architecture led to a 68% reduction in power consumption when compared to a traditional cloud data center architecture on average.

T24: Use energy-efficient hardware. The emissions of ML are related to used hardware. This is why using energy-efficient hardware to run ML models can reduce their power consumption. Energy-efficient hardware can include low-energy components. For example, the Tensor Processing Units (TPUs) developed by Google are seen as an energy-efficient alternative to CPUs and GPUs [25]. T25: Use power capping. Power capping is a technique used to limit the amount of power consumed by a device or system, such as a CPU, GPU, or server. It involves setting a maximum power consumption threshold for a device, and dynamically adjusting the power usage to ensure that it stays below that threshold. This is typically done to manage the power consumption and heat dissipa- tion of a device, and to prevent it from exceeding the power budget of a data center or other power-limited environment. Restricting the use of GPU resources can lead to reduced performance and longer execution times, but in certain configurations, it can also result in a significant reduction in energy consumption (up to 33%) with a moderate impact on performance [29].

T26: Use energy-aware scheduling. Energy-aware scheduling refers to a strategy that optimizes the scheduling of ML tasks. It dynamically schedules tasks or processes based on the current energy requirements and system conditions. The objective of an energy-aware dynamic scheduling policy is to make efficient use of available computational resources while still meeting energy budgets [52].

T27: Minimize referencing to data. ML models require read- ing and writing enormous amounts of data in the ML workflow. Reading data means retrieving information from storage, while writing data means storing or updating the information. These operations may increase unnecessary data movements and mem- ory usage, which influence the energy consumption of computing. To avoid non-essential referencing of data, data read and write operations must be designed carefully [48].

4.6 Management

The management category includes tactics for managing the ML- enabled system and ML model after their deployment. Only three tactics (T28-T30) fall under this category and are shown in Table 6.

Table 6: Green Tactics Related to Management

Tactic	Description	Target QA	Source
T28: Use informed adaptation	Adapt the model based on informed concept shift	Energy efficiency	[42]
T29: Retrain the model if needed	In case of concept shift, retrain the existing ML model instead of building a new one	Accuracy*	[42]
T30: Monitor computing power	Monitor computing power of an ML model in the long-term	Energy efficiency	[10][30]

The * means energy efficiency was considered a secondary QA

T29: Retrain the model if needed. Retraining a model refers to the process of updating or modifying an existing ML model. In the long term, concept drift may affect the accuracy of existing ML models. Retraining the model, by for example training it again with new data, is better than building it again from scratch in terms of sustainability [42].

T30: Monitor computing power. Estimating and calculating the energy footprint of an ML model can help to reduce its compu- tational power consumption. Monitoring the energy consumption of an ML model over the long term helps to identify those compo- nents where energy is being inefficiently utilized. This can serve as a starting point for making improvements to reduce energy con- sumption. There has been a lack of easy-to-use tools to do that, but recently

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

researchers have provided frameworks on how to estimate or calculate the energy footprint of ML-enabled systems [10, 30].

4.7 Targeted Quality Attributes

The majority of the tactics (21 out of 30) unsurprisingly aim to improve energy efficiency as their primary QA. Energy efficiency was mostly measured by the energy consumption savings achieved with the tactic. However, there were also some tactics that aimed primarily at other QAs, with energy efficiency being improved as a side effect. The description of these QAs and the number of tactics targeting them can be found in Table 7. One example was performance (3 out of 30), which was usually related to the time or throughput related to model training or inference. If the runtime or computational intensity can be reduced, this sometimes also influences energy efficiency positively. Additionally, some tactics had accuracy as their primary QA (3). While energy efficiency and accuracy are often regarded as a trade-off [66], our collection contains several tactics that try to increase both simultaneously. Lastly, recoverability (2) and resource utilization (1) appeared for a small number of tactics.

Overall, 17 tactics (T1-T6, T8, T10, T12-T15, T17, T22, T23, T25 and T26) were evaluated in experimental settings to provide ev- idence for their impact on these QAs. Most of the papers also provided evaluations of possible trade-offs with other QAs. These trade-offs include, for example, accuracy and latency. The other remaining tactics (13) were more along the lines of experience-based suggestions to improve QAs and environmental sustainability, with- out rigorous evaluations. While the provided argumentation was convincing, future work needs to provide empirical evidence to quantify the impact.

4.8 Scope of Architectural Tactics

The majority of tactics in our collection (20 of 30) are associated with low-level phases of the ML development life cycle, namely T28: Use informed adaptation. ML models may experience drift that affects their functionality. In these cases, the models must be adapted to deal with the drift. Informed adaptation refers to a method of adapting the ML model only when drift is detected. There- fore, the frequency of adaptation is smaller than in blind, periodic adaptation. Informed adaptation reduces unnecessary adaptations, which consequently saves energy [42].

data collection and processing, algorithm design, model optimiza- tion, and model training. In essence, these tactics are targeted at improving model quality instead of system quality. For example, tactics like Reduce the number of data features (T3) or Decrease model complexity (T8) could also be applied without a complete software system, i.e., for the training of a single ML model that is never integrated into a larger system. However, when these tactics are applied systematically at scale and continuously, they have strong architectural implications and significance. For example, although accuracy could be considered a model quality concern, the system actions taken in response to reduced accuracy as a trade-off for energy efficiency are architectural in nature. Moreover, some tac- tics from the early life cycle phases have a profound influence on architectural elements in the system, e.g., Consider reinforcement learning for energy efficiency (T9). For the categories deployment and management, the architectural significance of the tactics is more obvious, e.g., for Consider federated learning (T21), Apply cloud fog network architecture (T23), or Use informed adaptation (T28). All in all, the collection of tactics provides practitioners with holistic, architecture-centric means to improve the environmental sustainability of ML-enabled systems in all life cycle phases.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Table 7: Target Quality Attributes of the 30 Green Tactics

QA	Description	#
Energy efficiency	The ability to accomplish a task while minimizing energy consumption	21
Performance	The efficiency with which a task is achieved (e.g., speed, stability)	3
Accuracy	The level of how accurately the algorithm performs specified tasks.	3
Recoverability	The ability to restore and resume normal operations after a failure	2
Resource utilization	The ability to use and allocate resources efficiently	1

V. DISCUSSION

The collection of green architectural tactics obtained from our study provides guidance for architecting environmentally sustainable ML- enabled systems, and, as such, realize the societal promise that is expected of ML: minimizing its energy footprint. In the following, we report the main observations and reflections about our study.

Due to their diversity, tactic selection requires domain- specific expertise. As mentioned earlier, the tactics in Fig. 1 can- not be generalized to all use cases. For instance, some tactics are suitable for a specific algorithm type (e.g., Consider graph substitu- tion (T13) is only applicable to neural networks), while others are conditional to specific requirements (e.g., Consider transfer learn- ing (T16) requires a pre-trained model). Therefore, our catalog of tactics should not be interpreted as strict rules, but rather as rec- ommendations or available techniques for the architecture design of energy-efficient ML-enabled systems. In perspective, we believe that it would be beneficial for ML practitioners to have a catalog of tactics for specific use cases, such as using deep learning and its software architecture implications.

Most tactics are model-related rather than focusing on the full architecture of ML-enabled systems. An important obser- vation also discussed in the focus group is that most tactics found in our study focus on the ML model rather than the architecture of ML-enabled systems, i.e., a specific component rather than substan- tial parts of the architecture. We argue that this is because the field is still maturing, and reusable architecture knowledge about ML- enabled systems is still in the making. Furthermore, a related open problem is being able to separate the energy efficiency of model that, in many cases, energy consumption of ML models can be reduced without substantial reduction in accuracy [12, 54, 65, 70]. In general, given that energy consumption has become a major concern only recently and that ML-enabled systems are extremely energy demanding, we argue that future research should investigate possible trade-offs between energy efficiency and other QAs. Ana- lyzing interactions between multiple QAs could provide important insights into the design of ML-enabled systems.

VI. CONCLUSIONS AND NEXT STEPS

This paper provides a catalog of 30 green architectural tactics for ML-enabled systems organized in 6 categories, namely data-centric, algorithm design, model optimization, model training, deployment, and management. The tactics represent available techniques for designing energy-efficient ML-enabled systems. We integrated the collection into the Archive of Awesome and Dark Tactics.3 For transparency and reusability, we also provide a Zenodo repository.4

Despite the growing understanding of the environmental im- pacts of ML, there is still no consensus on how to best achieve sustainability. Our study serves as a starting point for further re- search about green architectural tactics for ML-enabled systems. Future research is necessary to validate and extend the results of this study, and to explore more generalized tactics applicable to differ- ent ML algorithms and ML-enabled system concerns. Furthermore, more

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

research is required to evaluate the effectiveness of several green architectural tactics in practice. In line with this, we plan to re-engineer existing open-source ML-enabled systems based on the tactics and to measure energy efficiency to compare the original versus the modernized system. As an alternative, we may conduct a case study where practitioners develop an ML-enabled system using the tactics catalog. Based on this, the usage of concrete tactics is analyzed, experiences are documented, and the catalog is refined. Finally, with the energy crisis and the explosion of ML applications in all sectors, energy efficiency is gaining traction as an impor- tant QA. Accordingly, it is important to create tactics dedicated to raising awareness of the energy footprint of such systems.

ACKNOWLEDGMENTS

We kindly thank our three focus group experts for their valuable time and feedback! Additionally, we thank Iffat Fatima (VU Ams- terdam) for her support with integrating the tactics into the AADT. Lewis and Ozkaya's work was supported by the Department of De- fense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center (DM23-2363). Muccini's work was supported by the PNRR ICSC National Re- search Centre for High Performance Computing, Big Data and Quantum Computing (CN00000013), under the NRRP MUR program funded by the NextGenerationEU. This research was supported by ExtremeXP, a project co-funded by the European Union Horizon Programme under Grant Agreement No. 101093164.

REFERENCES

[1] Brunno Abreu, Mateus Grellert, and Sergio Bampi. 2022. A framework for de- signing power-efficient inference accelerators in tree-based learning applications. Engineering Applications of Artificial Intelligence 109 (2022), 104638.
[2] Brunno A. Abreu, Mateus Grellert, and Sergio Bampi. 2020. VLSI Design of Tree- Based Inference for Low-Power Learning Applications. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 1–5. https://doi.org/10.1109/ ISCAS45731.2020.9180704
[3] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE, Montreal, QC, Canada, 291–300.
[4] Phyllis Ang, Bhuwan Dhingra, and Lisa Wu Wills. 2022. Characterizing the Efficiency vs. Accuracy Trade-off for Long-Context NLP Models. In 1st Workshop on Efficient Benchmarking in Nlp. Association for Computational Linguistics, Dublin, Ireland, 113—-121.
[5] Phyllis Ang, Bhuwan Dhingra, and Lisa Wu Wills. 2022. Characterizing the Efficiency vs. Accuracy Trade-off for Long-Context NLP Models. In Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP. Association for Computational Linguistics, Dublin, Ireland, 113–121. https://aclanthology. org/2022.nlppower-1.12
[6] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Car- bontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. arXiv:2007.03051.
[7] Len Bass, Paul Clements, and Rick Kazman. 2021. Software architecture in practice. Addison-Wesley Professional, Westford, MA, USA.
[8] Rodrigo F Berriel, Andre Teixeira Lopes, Alexandre Rodrigues, Flavio Miguel Varejao, and Thiago Oliveira-Santos. 2017. Monthly energy consumption fore- cast: A deep learning approach. In 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, Anchorage, AK, USA, 4283–4290.
[9] Glenn A Bowen. 2009. Document analysis as a qualitative research method.
Qualitative research journal 9, 2 (2009), 27–40.
[10] Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2021. IrEne: Interpretable Energy Prediction for Transformers. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, online, 2145–2157. https://par.nsf.gov/biblio/10297211

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

[11] Katerina Chinnappan, Ivano Malavolta, Grace A. Lewis, Michel Albonico, and Patricia Lago. 2021. Architectural Tactics for Energy-Aware Robotics Software: A Preliminary Study. In Software Architecture, Stefan Biffi, Elena Navarro, Welf Löwe, Marjan Sirjani, Raffaela Mirandola, and Danny Weyns (Eds.). Vol. 12857. Springer International Publishing, Cham, 164–171. https://link.springer.com/10. 1007/978-3-030-86044-8_11 Series Title: Lecture Notes in Computer Science.

[12] Santiago del Rey, Silverio Martínez-Fernández, Luís Cruz, and Xavier Franch. 2023. Do DL Models and Training Environments Have an Impact on Energy Consumption?. In 49th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA). IEEE, 1–8.

[13] Priyadarshan Dhabe, Param Mirani, Rahul Chugwani, and Sadanand Gandewar. 2021. Data Set Reduction to Improve Computing Efficiency and Energy Consump- tion in Healthcare Domain. In Digital Literacy and Socio-Cultural Acceptance of ICT in Developing Countries. Springer, 53–64.

[14] Issam El Naqa and Martin J. Murphy. 2015. What Is Machine Learning? Springer International Publishing, Cham, 3–11.

[15] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston, MA, USA.

[16] Eva García-Martín, Niklas Lavesson, Håkan Grahn, Emiliano Casalicchio, and Veselka Boeva. 2021. Energy-Aware Very Fast Decision Tree. Int. J. Data Sci. Anal. 11, 2 (March 2021), 105–126.

[17] Bahar Gezici and Ayça Kolukısa Tarhan. 2022. Systematic literature review on software quality for AI-based software. Empirical Software Engineering 27, 3 (2022), 66.

[18] Neil B. Harrison and Paris Avgeriou. 2010. How do architecture patterns and tactics interact? A model and annotation. Journal of Systems and Software 83, 10 (Oct. 2010), 1735–1758. https://doi.org/10.1016/j.jss.2010.04.067

[19] Lukas Heiland, Marius Hauser, and Justus Bogner. 2023. Design Patterns for AI- based Systems: A Multivocal Literature Review and Pattern Repository. In 2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN). IEEE, Melbourne, Australia, 184–196. https://doi.org/10.1109/ CAIN58948.2023.00035

[20] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. Journal of Machine Learning Research 21, 248 (2020), 1–43. http://jmlr.org/papers/v21/20-312.html

[21] Jennifer Horkoff. 2019. Non-functional requirements for machine learning: Challenges and new directions. In 2019 IEEE 27th International Requirements Engineering Conference (RE). IEEE, 386–391.

[22] Anton Jansen and Jan Bosch. 2005. Software Architecture as a Set of Architectural Design Decisions. In 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05), Vol. 2005. IEEE, 109–120. https://doi.org/10.1109/WICSA.2005.61

[23] Nitthilan Kanappan Jayakodi, Syrine Belakaria, Aryan Deshwal, and Janard- han Rao Doppa. 2020. Design and optimization of energy-accuracy tradeoff networks for mobile platforms via pretrained deep models. ACM Transactions on Embedded Computing Systems (TECS) 19, 1 (2020), 1–24.

[24] Mladan Jovanovic and Mark Campbell. 2022. Generative Artificial Intelligence: Trends and Prospects. Computer 55, 10 (Oct. 2022), 107–112. https://doi.org/10. 1109/MC.2022.3192720

[25] Lynn H Kaack, Priya L Donti, Emma Strubell, George Kamiya, Felix Creutzig, and David Rolnick. 2022. Aligning Artificial Intelligence with Climate Change Mitigation. Nature Climate Change 12, 6 (2022), 518–527.

[26] Minsu Kim, Walid Saad, Mohammad Mozaffari, and Merouane Debbah. 2021. On the Tradeoff between Energy, Precision, and Accuracy in Federated Quantized Neural Networks. In ICC 2022 - IEEE International Conference on Communications. IEEE, 2194–2199.

[27] Young Geun Kim and Carole-Jean Wu. 2020. Autoscale: Energy efficiency opti- mization for stochastic edge inference using reinforcement learning. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 1082–1096.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

[28] Jyrki Kontio, Johanna Bragge, and Laura Lehtola. 2008. The Focus Group Method as an Empirical Tool in Software Engineering. In Guide to Advanced Empirical Software Engineering, Forrest Shull, Janice Singer, and Dag I. K. Sjøberg (Eds.).
Springer London, London, 93–116. https://doi.org/10.1007/978-1-84800-044-5_4
[29] Adam Krzywaniak, Pawel Czarnul, and Jerzy Proficz. 2022. GPU Power Capping for Energy-Performance Trade-Offs in Training of Deep Convolutional Neural Networks for Image Recognition. In International Conference on Computational Science. Springer, 667–681.
[30] Mohit Kumar, Xingzhou Zhang, Liangkai Liu, Yifan Wang, and Weisong Shi. 2020. Energy-Efficient Machine Learning on the Edges. In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 912– 921.
[31] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 (2019), 1–8.
[32] Patricia Lago. 2022. Awesome Tactic Template. Digital Sustainability Center (DiSC). https://s2group.cs.vu.nl/AwesomeAndDarkTactics/tactics/2022-01-01- awesome-template
[33] Grace A Lewis, Ipek Ozkaya, and Xiwei Xu. 2021. Software architecture challenges for ml systems. In 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 634–638.
[34] Ivano Malavolta, Katerina Chinnappan, Stan Swanborn, Grace A. Lewis, and Patricia Lago. 2021. Mining the ROS ecosystem for Green Architectural Tactics in Robotics and an Empirical Evaluation. In 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). IEEE, Madrid, Spain, 300–311. https://ieeexplore.ieee.org/document/9463098/
[35] Susmita Dey Manasi, Farhana Sharmin Snigdha, and Sachin S Sapatnekar. 2020. NeuPart: Using analytical models to drive energy-efficient partitioning of CNN computations on cloud-connected mobile clients. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 28, 8 (2020), 1844–1857.
[36] Matthew B. Miles. 2014. Ch. 4 - Fundamentals of qualitative data analysis. In Qualitative data analysis: A methods sourcebook (3rd ed.), Matthew B. Miles and A. Michael Huberman (Eds.). Sage Publications, 56–89.
[37] Thaha Mohammed, Aiiad Albeshri, Iyad Katib, and Rashid Mehmood. 2020. UbiPriSEQ—Deep Reinforcement Learning to Manage Privacy, Security, Energy, and QoS in 5G IoT HetNets. Appl. Sci. 10, 20 (Oct. 2020), 7120.
[38] Elena Morotti, Davide Evangelista, and Elena Loli Piccolomini. 2021. A green prospective for learned post-processing in sparse-view tomographic reconstruc- tion. Journal of Imaging 7, 8 (2021), 139.
[39] Henry Muccini and Karthik Vaidhyanathan. 2021. Software architecture for ML- based systems: What exists and what lies ahead. In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN). IEEE, 121–128.
[40] Gastón Márquez, Hernán Astudillo, and Rick Kazman. 2023. Architectural tactics in software architecture: A systematic mapping study. Journal of Systems and Software 197 (March 2023), 111558. https://doi.org/10.1016/j.jss.2022.111558
[41] Harikumar Pallathadka, Malik Mustafa, Domenic T Sanchez, Guna Sekhar Sajja, Sanjeev Gour, and Mohd Naved. 2023. Impact of machine learning on man- agement, healthcare and agriculture. Materials Today: Proceedings 80 (2023), 2803–2806.
[42] Lorena Poenaru-Olaru, June Sallou, Luis Cruz, Jan S. Rellermeyer, and Arie van Deursen. 2023. Retrain AI Systems Responsibly! Use Sustainable Concept Drift Adaptation Techniques. In 2023 IEEE/ACM 7th International Workshop on Green And Sustainable Software (GREENS). IEEE, 17–18. https://doi.org/10.1109/GREENS59328.2023.00009
[43] Lena Pons and Ipek Ozkaya. 2019. Priority quality attributes for engineering ai-enabled systems. arXiv preprint arXiv:1911.02912 (2019), 1–4.
[44] Giuseppe Procaccianti, Patricia Lago, and Grace A. Lewis. 2014. A Catalogue of Green Architectural Tactics for the Cloud. In 2014 IEEE 8th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems. IEEE, Victoria, BC, Canada, 29–36. https://doi.org/10.1109/MESOCA. 2014.12

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

[45] Bita Darvish Rouhani, Azalia Mirhoseini, and Farinaz Koushanfar. 2016. DeLight: Adding Energy Dimension To Deep Neural Networks. In ISLPED '16: Proceed- ings of the 2016 International Symposium on Low Power Electronics and Design. Association for Computing Machinery, New York, NY, USA, 112–117.
[46] Iqbal H Sarker. 2021. Machine learning: Algorithms, real-world applications and research directions. SN computer science 2, 3 (2021), 160.
[47] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI.Commun. ACM 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831
[48] Shriram Shanbhag, Sridhar Chimalakonda, Vibhu Saujanya Sharma, and Vikrant Kaulgud. 2022. Towards a Catalog of Energy Patterns in Deep Learning Develop- ment. In The International Conference on Evaluation and Assessment in Software Engineering 2022. ACM, Gothenburg Sweden, 150–159. https://doi.org/10.1145/3530019.3530035
[49] Julien Siebert, Lisa Joeckel, Jens Heidrich, Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, and Mikio Aoyama. 2020. Towards guidelines for assessing qualities of machine learning systems. In Quality of Information and Communications Technology: 13th International Conference, QUATIC 2020, Faro, Portugal, September 9–11, 2020, Proceedings 13. Springer, 17–31.
[50] Martino Sorbaro, Qian Liu, Massimo Bortone, and Sadique Sheik. 2020. Opti- mizing the energy consumption of spiking neural networks for neuromorphic applications. Frontiers in neuroscience 14 (2020), 662.
[51] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 3645–3650. https://doi.org/10.18653/ v1/P19-1355
[52] Yuxuan Sun, Sheng Zhou, and Deniz Gündüz. 2020. Energy-aware analog ag- gregation for federated learning with redundant data. In ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 1–7.
[53] Aimee Van Wynsberghe. 2021. Sustainable AI: AI for Sustainability and the Sustainability of AI. AI and Ethics 1, 3 (2021), 213–218.
[54] Roberto Verdecchia, Lus Cruz, June Sallou, Michelle Lin, James Wickenden, and Estelle Hotellier. 2022. Data-Centric Green AI: An Exploratory Empirical Study. In 2022 International Conference on ICT for Sustainability (ICT4S). IEEE, 35–45.
[55] Roberto Verdecchia, Emelie Engström, Patricia Lago, Per Runeson, and Qunying Song. 2023. Threats to Validity in Software Engineering Research: A Critical Reflection. Information and Software Technology 164 (Sept. 2023), 107329. https: //doi.org/10.1016/j.infsof.2023.107329
[56] Roberto Verdecchia, June Sallou, and Luis Cruz. 2023. A Systematic Review of Green AI. WIREs Data Mining and Knowledge Discovery 13, 4 (July 2023), e1507. https://doi.org/10.1002/widm.1507
[57] Andreas Vogelsang and Markus Borg. 2019. Requirements engineering for ma- chine learning: Perspectives from data scientists. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW). IEEE, 245–251.
[58] Sophie Vos, Patricia Lago, Roberto Verdecchia, and Ilja Heitlager. 2022. Archi- tectural Tactics to Optimize Software for Energy Efficiency in the Public Cloud. In 2022 International Conference on ICT for Sustainability (ICT4S). IEEE, Plovdiv,Bulgaria, 77–87. https://ieeexplore.ieee.org/document/9830087/
[59] Qu Wang, Yong Xiao, Huixiang Zhu, Zijian Sun, Yingyu Li, and Xiaohu Ge. 2021. Towards Energy-efficient Federated Edge Intelligence for IoT Networks. In 2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, 55–62.
[60] Yu Wang, Rong Ge, and Shuang Qiu. 2020. Energy-Aware DNN Graph Optimiza- tion. In Resource-Constrained Machine Learning (ReCoML) Workshop of MLSys 2020 Conference. MLSys, Austin, TX, USA., 1–7.
[61] Yue Wang, Ziyu Jiang, Xiaohan Chen, Pengfei Xu, Yang Zhao, Yingyan Lin, and Zhangyang Wang. 2019. E2-Train: Training State-of-the-art CNNs with Over 80% Energy Savings. In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/ 2019/file/663772ea088360f95bac3dc7ffb841be-Paper.pdf

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

[62] Hironori Washizaki, Foutse Khomh, Yann-Gaël Guéhéneuc, Hironori Takeuchi, Naotake Natori, Takuo Doi, and Satoshi Okuda. 2022. Software-engineering design patterns for machine learning applications. Computer 55, 3 (2022), 30–39.

[63] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. Pro- ceedings of Machine Learning and Systems 4 (2022), 795–813.

[64] Jinsong Wu, Song Guo, Jie Li, and Deze Zeng. 2016. Big data meet green chal- lenges: Greening big data. IEEE Systems Journal 10, 3 (2016), 873–887.

[65] Xu, Yinlena, Martínez-Fernández, Silverio, Martinez, Matias, and Franch Xavier. 2023. Energy Efficiency of Training Neural Network Architectures: An Empirical Study. In Hawaii International Conference on System Sciences (HICSS). University of Hawai'i, 781–790. https://hdl.handle.net/10125/102727

[66] Haichuan Yang, Yuhao Zhu, and Ji Liu. 2019. Energy-constrained compression for deep neural networks via weighted sparse projection and layer input masking. In International Conference on Learning Representations (ICLR). OpenReview, 1–20.

[67] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. 2017. Designing Energy-Efficient Convolutional Neural Networks Using Energy-Aware Pruning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 5687–5695.

[68] Xiangyu Yang, Sheng Hua, Yuanming Shi, Hao Wang, Jun Zhang, and Khaled B. Letaief. 2020. Sparse Optimization for Green Edge AI Inference. Journal of Communications and Information Networks 5, 1 (2020), 1–15.

[69] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. 2020. Energy efficient federated learning over wireless commu- nication networks. IEEE Transactions on Wireless Communications 20, 3 (2020), 1935–1949.

[70] Tim Yarally, Luıs Cruz, Daniel Feitosa, June Sallou, and Arie Van Deursen. 2023. 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). IEEE, Melbourne, Australia, 25–36. https://doi.org/10.1109/CAIN58948.2023.00012

[71] Barzan A Yosuf, Sanaa H Mohamed, Mohammed M Alenazi, Taisir EH El-Gorashi, and Jaafar MH Elmirghani. 2021. Energy-Efficient AI over a Virtualized Cloud Fog Network. In Proceedings of the Twelfth ACM International Conference on Future Energy Systems. ACM, 328–334