Sustainability of Edge AI at Scale

An empirical study on the sustainability of Edge AI in terms of energy consumption.

Master Thesis

Rover van der Noort

Delft University of Technology

TU Delft

Sustainability of Edge AI at Scale

An empirical study on the sustainability of Edge AI in terms of energy consumption.

by

Rover van der Noort

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday May 8, 2024 at 15:00.

Student number:	4680502		
Project duration:	September 20, 2023 – May 8, 2024		
Thesis committee:	Luis Cruz,	TU Delft	daily supervisor
	Prof. Arie van Deursen,	TU Delft	thesis advisor
	Ujwal Gadiraju	TU Delft	external advisor
	Silverio Martínez-Fernández	UPC BarcelonaTech	external supervisor

This thesis is confidential and cannot be made public until May 8, 2024.

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

Edge AI is an architectural deployment tactic that brings AI models closer to the user and data, relieving internet bandwidth usage and providing low latency and privacy. It remains unclear how this tactic performs at scale, since the distribution overhead could impact the total energy consumption. We identify four architectural scalability factors that could impact the energy consumption of AI: environment, optimisation, throughput, and overhead. The latter consists of downloading, verification, and updating the model over time. This work performs an empirical study on the sustainability of Edge AI compared to Cloud AI at scale in terms of energy consumption. For the environment variable, energy consumption measurement experiments are run on a cloud device and multiple edge devices, various quantized models for optimisation, and various throughput levels per hour. We simulate the distribution overhead and combine the results with the measurements to find the holistic energy efficiency of each architectural strategy. We find that all four variables impact energy consumption, but the main contributors are environment, throughput, and overhead. We observe that Edge AI is most energy-efficient in low-distribution, low-demand scenarios, whereas in high-distribution, high-demand scenarios Cloud AI is better optimised and outperforms Edge AI in energy efficiency. This means that developers depending on their use case and the project's scalability need to consider these quality attributes for the most sustainable architectural solution.

Preface

To whom it may concern:

This is (for now) my final contribution to the academic field as a Master's student in the Software Engineering Research Group (SERG) of Delft University of Technology. Starting in 2017 with my Bachelor's degree, I couldn't have phanthomed the knowledge and capabilities I gained over the years, which resulted in this contribution.

Although I've always had an interest in nature, sustainability, and for instance recycling, my passion for Computer Science and Engineering (CSE) outweighed these preferences and I started my studies in Delft. My ambition to contribute to the climate challenges subsided a bit as generally, CSE focuses on accuracy over sustainability.

This changed when I took the course Sustainable Software Engineering taught by my daily supervisor for this thesis, *Luis Cruz*, and became aware of this field. For this course, we created an energy measurement library for the PyTorch library to gain insight into the specific energy consumption within a model and create more awareness for Green AI[1]. This combined my interest and I discovered the huge potential of finding technical solutions to the carbon footprints of IT in the world.

This led to the cooperation that constituted this thesis in the field of Green AI. Over the span of 9 months starting in September 2023, together with my other daily supervisor, *Silverio Martínez-Fernández* from UPC BarcelonaTech, I worked on the lack of awareness of the sustainability of Edge AI at scale in terms of energy consumption.

I want to thank Luis and Silverio who provided me with weekly support and feedback and made this thesis possible. We had many constructive yet critical meetings that allowed me to finish it with novel contributions and on time. Furthermore, I want to thank *Arie van Deursen* and *Ujwal Gadiraju* for taking time out of their busy schedules to grade my graduation. Lastly, I want to thank my friends and family for their unwavering support throughout the thesis and my complete time at the university.

Furthermore, even though I'm not continuing my academic career with a PhD at this time, I'm still dedicated to spending my time and capabilities in the field of sustainable software engineering and I urge other developers to consider their environmental impacts and act accordingly. Generally, software allows for many good things but needs correct control to be sustainable, otherwise, the cost might outweigh the benefits.

Rover van der Noort
Delft, April 2024

[1] See: `https://github.com/GreenAITorch/GATorch`

Contents

Abstract i

Preface ii

1 Introduction 1

2 Related Work 3
2.1 Green AI . . . 3
2.2 Cloud AI . . . 4
2.3 Fog AI . . . 6
2.4 Edge AI . . . 6
2.4.1 Green Edge AI . . . 8
2.4.2 Research Gap . . . 13

3 Experimental Design 14
3.1 Methodology . . . 14
3.1.1 Variables . . . 15
3.2 Experimental Setup . . . 17
3.2.1 Normalization . . . 19
3.2.2 Analysis strategy . . . 20
3.2.3 Replication package . . . 21

4 Results 22
4.1 Normality . . . 23
4.2 Environment (RQ1.1) . . . 24
4.2.1 Edge vs Cloud . . . 24
4.2.2 Optimal Edge device . . . 26
4.3 Quantization (RQ1.2) . . . 27
4.3.1 Non-quantized vs Quantized . . . 27
4.3.2 Optimal Quantization level . . . 30
4.4 Throughput (RQ1.3) . . . 34
4.5 Overhead (RQ1.4) . . . 36

5 Discussion 39
5.1 Implications . . . 39
5.2 Threads to validity . . . 40
5.3 Future work . . . 41

6 Conclusion 43

References 44

A Results 52

B Justification 53

1

Introduction

Artificial Intelligence (AI) has been in increasing demand and coupled with the increase in size and complexity of the models, this significantly impacts the energy consumption and carbon footprint in the world [31, 128]. The energy consumed in data centres to run these models is enormous and the resources and infrastructure needed to produce the required hardware are expensive [128]. Many AI models prioritise achieving maximum accuracy without considering resource constraints, a paradigm known as Red AI. In contrast, Green AI emphasises energy efficiency over accuracy [101].

AI development consists of multiple steps, such as data collection, experimentation, training, and deployment, which can all consume high resources. While most Green AI research has focused on the training phase of AI development, less attention has been paid to the energy costs associated with model inference once it is deployed. At the same time, this generally consumes more energy over the life-cycle of the model [128]. Therefore, a more holistic approach to the investigation of energy consumption of AI needs to be made [128, 129].

The inference costs can have a significant impact on the total energy consumption of the whole pipeline [26]. This is because some models have become so popular that many users infer many requests, which scales up the energy consumption. For instance, with the release of ChatGPT, many people started using AI that had never before, and even though OpenAI is optimising the energy efficiency of their processes, this can lead to a phenomenon called Jevons Paradox[1]. This states that improving the efficiency of a resource increases the demand leading to an increase in the use of the resource, instead of a reduction due to the efficiency. Even though the energy consumption of inference has remained constant due to optimisations, this increase in usage results in higher overall energy consumption [26]. It is therefore important to create more awareness about the energy consumption of AI for both the users and developers and reduce it where possible [56].

The majority of AI models are trained and deployed in cloud environments, consisting of centralised or distributed data centres consisting of High-Performance Computing (HPC) hardware configurations that are optimised for high-performance AI training and inference. These cloud instances are performing well in accuracy and inference speeds, however, their energy consumption and carbon footprint are inherently large. On the other hand, cloud instances are so large that smaller models might result in underutilisation of the resources and lost energy running idle [95, 128].

Cloud providers have been investing in better carbon awareness and tools that reduce the energy consumption of training and deployment of AI models by for instance scheduling [109, 116, 128, 129] or deferring requests to locations with cleaner energy [128, 129]. However, due to the proprietary nature of the cloud, it is hard to do direct investigations into the energy consumption of these data centres. The cloud providers charge their customers for all the usage of their resources, which can increase significantly once the model scales in the amount of inferences it performs. This incentivised some AI developers to consider Edge AI methods to reduce energy consumption and high monetary costs to cloud providers.

Edge AI is a deployment strategy characterised by running the AI on the devices where the data is located. An increasing number of edge devices are connected to the internet [2], ranging from mobile

[1] See: `https://en.wikipedia.org/wiki/Jevons_paradox`

phones to consumer laptops and specialised IoT devices. Depending on the application, Edge AI promises to bring the models closer to the data source in order to decrease latency. Smart grid [107, 109, 110], smart city [4, 59, 102, 134], IoT in Industry 4.0 [21, 32, 86, 88, 107, 118] and self-driving cars [61] are examples of use cases, which benefit from Edge AI's features. In practice, many organisations choose a hybrid strategy between cloud and edge for their deployment strategy [34, 35, 103]. The experiments of this study are performed in the context of Smaller Language Models (SLM), which are smaller versions of Large Language Models (LLM). These are popular models with a wide range of applications such as real-time applications like chatbots. Edge AI benefits from lower response times and offline usage [70] since the strategy moves the computation to the end user device.

Logically, smaller devices on the edge generally consume less energy than large Cloud HPC devices, however, they are less optimised and can significantly increase the processing duration per request. This means latency could increase and the maximum throughput is smaller per device, which means high-throughput applications require a more complex setup with multiple edge devices. Therefore, the sustainability of Edge AI is still up for debate [1], since the correlated overhead of such a network could introduce significant energy consumption. Because Edge AI can operate offline, this can reduce the communication bandwidth of the internet, however, Edge AI comes with a high level of duplication across a potentially wide range of devices. This means a compilated maintenance setup is required and thus the related overhead could impact the sustainability [128]. Little investigation into the energy consumption of deployment strategies between Edge AI and Cloud AI has been done. Researchers are working on studies that push for more sustainable research towards the edge like GreenEdge [44]. However, they lack the needed comparison between the edge and the cloud at scale. This raises the question of whether Edge AI including all these distribution overheads decreases energy consumption compared to the highly optimised cloud solutions.

An IT company that uses various AI applications might need to come up with a deployment strategy for their models that optimises energy and cost efficiency. For example, the company wants to deploy a coding assistant model for their developers that runs on their internal codebase and has a very high utility and update rate. The company can run this on a cloud instance or deploy the model on the developers' workstations. Another scenario for the same company entails a support chatbot for their clients. This support chatbot can help the user by providing detailed and relevant answers about the company's software but it is generally only used for a few questions a day. The other option is to deploy on the cloud or on-premise with the client. It remains unclear for this company what the best strategy is for either use case. Is Edge AI an energy-efficient way to deploy these models and which factors determine this?

To find recommendations for AI developers about the sustainability of various AI deployment factors, this thesis aims **to investigate the energy consumption of different AI deployment strategies**, with a focus on understanding the **scalability factors** influencing energy efficiency in Edge AI compared to Cloud AI.

Document Structure

This document starts by introducing the necessary related work in the field of Green AI and Edge AI. Our investigation of existing literature reveals a gap in understanding the scalability of Edge AI concerning its energy consumption and carbon footprint, motivating the need for further research in this area.

In response to the identified gap, this study aims to investigate the impact of four scalability factors; environment, quantization, throughput, and model lifespan on energy consumption in Edge AI deployment strategies. We identify the relevant variables, followed by a discussion of the experimental setup to measure the appropriate energy consumption values and how to compare them. We conduct empirical experiments to quantify the energy consumption of various devices across different variables. Additionally, we develop a simulation model to assess the energy consumption of the extended lifespan of AI models at scale.

The results show that all the identified scalability factors of Edge AI impact the overall energy consumption. We observe that the throughput and overhead factors are the main contributors to this difference. Therefore, Edge AI can only be energy efficiently applied in a low-demand, low-throughput environment with specialised investigation for device and optimisation strategy. For AI applications at scale, the energy efficiency on the cloud is better optimised. AI developers need to thoroughly investigate their most efficient deployment strategy for each use case.

2

Related Work

This section describes the previous work in the fields of Green AI, Cloud AI and Edge AI. First, we explore the notion of Green AI and the relevant research from the last few years. This is followed by the identification of the pros and cons of both Cloud AI and Edge AI and their sustainability efforts. Lastly, we look at techniques for measuring energy consumption and finally, we present the identified research gap that this study aims to fill.

2.1. Green AI

In 2019, Schwartz et al. [101] introduced the notion of Red and Green AI, which respectively stand for accuracy-focused and energy-focussed AI development. This paper established the current research field of sustainable software engineering for AI and this shows the infancy of this research field.

Luccioni et al. [72] looked into the carbon emissions of ML models over time and discovered that they have increased. They concluded that higher energy consumption and carbon emissions do not correlate with higher accuracy. Another study by Luccioni et al. [73] investigated the carbon footprint of BLOOM, a 176B parameter model, and found that its training emitted 25-50 tonnes of CO_2eq, which is roughly equivalent to 6-12 passenger cars driving for a year. They recommended further research into the energy consumption of the inference step in the AI pipeline. Desisilavov et al. [26] observed an increasing trend in AI energy consumption as well, however, the hardware and software optimisations have decreased the expected growth in consumption, although it is still increasing due to the general higher usage of AI. Castano et al. [16] observed decreased carbon reporting on Hugginface for which they proposed a method to create more awareness about the footprints of these models. Furthermore, they found a correlation between carbon emissions and model and dataset sizes. Zhou et al. [137] proposed HULK, an energy efficiency benchmark platform in which they report training and inference times of a selection of LLMs and the associated costs.

Saheb et al. [96] reviewed AI for sustainable energy and found that AI optimizations are a relevant part towards more sustainable AI use. Chien et al. [20] proposed a CarbonMin optimisation that shifts the workload geographically in order to reduce carbon emissions from the request. The authors found that response latency is weakly correlated with user location, so in case of a bad connection, this could negatively impact the user experience.

Martinez et al. [75, 95, 130] found that the model architecture has an effect on energy consumption and that the training environment should factor in the model architecture for most energy-efficient training by optimising the GPU utilisation. Li et al. [65] found that the hardware used for training has an impact on energy consumption and that training on GPU-enabled devices is more energy efficient than CPU-only devices. Yarally et al. [132] investigated the energy efficiency of various hyperparameters optimization techniques and found that Bayesian optimisation is the most efficient. Moreover, they found that convolutional layers in a CNN are most power-hungry, but that the complexity can often be reduced for more energy efficiency without much loss in accuracy. Another study found that batch size significantly impacts the energy consumption and inference speed of the models [131].

Verdecchia et al. [120] investigated a data-centric approach to reduce the energy consumption of AI systems and found that some dataset operations can significantly reduce energy consumption without

Green Architectural Tactics for ML-Enabled Systems

Data-centric	Algorithm design	Model optimization	Model training	Deployment	Management
T1: Apply sampling techniques **T2**: Remove redundant data **T3:** Reduce number of data features **T4:** Use input quantization **T5:** Use data projection	**T6:** Choose an energy-efficient algorithm **T7**: Choose a lightweight algorithm alternative **T8:** Decrease model complexity **T9:** Consider reinforcement learning for energy efficiency **T10:** Use dynamic parameter adaptation **T11:** Use built-in library functions*	**T12:** Set energy consumption as a model constraint **T13:** Consider graph substitution **T14:** Enhance model sparsity **T15:** Consider energy-aware pruning **T16**: Consider transfer learning **T17:** Consider knowledge distillation	**T18:** Use quantization-aware training **T19:** Use checkpoints during training **T20:** Design for memory constraints*	**T21:** Consider federated learning **T22:** Use computation partitioning **T23:** Apply cloud fog network architecture **T24:** Use energy-efficient hardware **T25:** Use power capping **T26:** Use energy-aware scheduling **T27**: Minimize referencing to data*	**T28:** Use informed adaptation* **T29:** Retrain the model if needed **T30:** Monitor computing power

The symbol * means the tactic was found with the help of the focus group.

Figure 2.1: Catalog of Green AI techniques [56].

reducing accuracy. The same authors [119] also provide a systematic literature review of Green AI. They found that Edge Computing is a hot topic, while deployment is an under-considered phase. This shows the need for more research into the energy consumption of AI inferences, as this contributes largely to the overall energy consumption [127].

This overview of the latest developments in Green AI shows the potential of reducing the carbon footprint of AI models by all kinds of techniques as shown in the synthesis in Figure 2.1 [56]. All these papers have shown the broad possibilities of energy consumption reduction for the complete AI development field. However, they also show the difficulty in measuring and reporting accurate results. Moreover, we find a gap in research on the scalability of these systems and the effect on the energy consumption of AI at scale.

2.2. Cloud AI

Cloud computing uses large data centres full of HPC hardware to execute AI applications. There are many options for cloud deployment and you can configure the hardware configuration to scale up and down based on your usage. You are charged by the cloud providers for the usage of their systems, usually on a per-request basis. Extensive use of those systems can become quite expensive [38].

The simplest cloud configuration for AI deployment consists of a single model in a virtual machine or container environment in a specific region with a specific hardware set. However, deployment engineers can theoretically scale the system infinitely. In Table 2.1, we show an identification of the advantages and disadvantages of Cloud AI and how often they were mentioned by literature. Due to the *centralised* nature and high *configurability* of cloud deployment, it is the preferred deployment option for almost all applications nowadays. It offers a *reliable* system that can *scale*, and cloud providers are improving the *sustainability* of the data centre facilities, like using renewable energy. However, these data centres still have high *energy consumption* for executing the requests, high *internet transmission* energy costs, and high *embodied carbon costs* due to the production of the HPC hardware. This all results in high *monetary costs* for the cloud users. Finally, there are *privacy* concerns about sending user data over the internet and aggregating this data on the cloud.

Devices

Over the years, the primary method of computation has shifted from mostly CPU loads for webservers and APIs to mostly GPU loads for AI training and inference [128]. Although the power consumption of GPUs is generally higher than that of CPUs, due to the parallel computational abilities, the energy consumption for AI loads does not necessarily have to be higher [80]. Li et al. [65] did a study on CNNs and measured their CPU and GPU energy consumption finding variability in network topology and batch size. Cheng et al. [19] found that for small operations, CPU-only outperforms GPU, but for more complex operations, the GPU or a CPU-GPU co-processing setup reduces energy consumption.

Moreover, Wang et al. [124] benchmarked the performance and energy efficiency of various AI accelerators on multiple GPU and Tensor Processing Unit (TPU) configurations. Ma et al. [74] propose

Table 2.1: Advantages and challenges of Cloud AI.

Advantages	**Citations**
Sustainability efforts	[11, 30, 77, 116, 128]
Scalability	[11, 68, 116, 128]
Redundancy and reliability	[11, 116, 127]
Centralised setup, easy maintenance	[11, 116]
Offers wide array of configurations	[11, 116]
High performance	[127]
High quality training	[127]
Challenges	
High energy consumption	[11, 30, 68, 69, 77, 89, 116]
Privacy concerns	[69, 128]
High monetary costs	[38, 137]
High embodied carbon costs	[69, 128]
Responsiveness (latency)	[32, 69]
High internet bandwidth usage	[32, 89]

a framework called GreenGPU, which distributes workloads dynamically based on the characteristics of the computation by which they can reduce energy consumption.

More specialised hardware for AI, such as Field Programmable Gate Arrays (FPGA), can outperform both CPU and GPU hardware configurations if model complexity increases [93], but this depends on memory localisation and therefore depends on the application [12] and the optimization technique [79]. Field Al-Ali et al. [3] found a significant increase in inference time using an FPGA, compared to a consumer-grade GPU for image processing. Boutros et al. [13] also found in their comparison of AI-optimised FPGAs and GPUs a significant compute speedup for the latter.

This shows the complexity of hardware configurations and their impact on the energy efficiency of deployment systems. Furthermore, specialised hardware also brings significant embodied carbon costs in their production. Furthermore, hardware components have increasingly higher power ratings, which is concerning for their sustainability, however, their computing capability is growing as well.

Providers

Escribano [30] performed an experiment where he compared the energy consumption of various cloud providers and found that cloud providers differ in total carbon emissions. It is hard to establish the accuracy of these results due to the complexity of cloud environments.

Large cloud providers such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure provide users with many options and tools to optimise or monitor performance and carbon footprint. They offer sustainability dashboards that give insights to their users on the carbon impact of their projects [40], which they calculate by estimating the carbon footprint over multiple scopes from fuel used by generators, energy consumption per region, to the carbon cost of the production of the hardware [41]. They offer development tools that reduce the carbon emissions of projects in the cloud. For instance, these methods refer requests to regions with cleaner energy [42] and they make recommendations for minimising unused cloud resources [43]. OpenStack is an Open-Source Software (OSS) platform that allows you to build a cloud environment for users to set up their nodes with configurations. They maintain a marketplace with tools such as environmental dashboards [113].

All these sustainability efforts are important to reduce the total energy consumption in various ways, however, it remains hard to determine what the actual impact is of these optimisation techniques: once they are inside these proprietary data centres, they are hard to measure. Therefore, more research is required into the actual energy consumption of cloud infrastructures.

Scalability

Only a few research papers look into the measured energy consumption of these cloud setups at scale and many studies rely only on estimates to come to some quantification [84]. However, it is hard to establish the complete carbon footprint of cloud deployment, since models in production can have varied life cycles and usages. The amount of throughput, the model size, and the hardware configuration

in the cloud can significantly impact the final power consumption of the model's deployment. Tuli et al. [116] proposed an AI holistic resource manager model to manage sustainable cloud computing.

Lin et al. [68] performed a study on an energy consumption measurement system for a multi-component cloud system at scale. They proposed Distributed Energy Meters (DEM) for heterogeneous cloud environments, outperforming state-of-the-art (SOTA) energy consumption estimation methods.

The internet traffic a cloud-deployed model generates and the associated costs are reasons AI deployers select Edge AI. As you pay for the bandwidth of your application, the monetary costs could drastically increase once the application scales up. Especially multimedia internet traffic, like photos or videos, can use more energy than textual traffic. However, the network traffic energy costs of the deployment strategies are often overlooked and depending on the actual utility of the model, this overhead can have a significant impact [5, 52].

2.3. Fog AI

Fog deployment is the strategy between cloud model deployment and edge deployment, which exists at various levels, ranging from cloud regions to local company networks. This strategy benefits from similar advantages as the edge although not completely, as the offline benefit trait does not apply to Fog, meaning it has a different set of challenges [11]. For instance, Shen et al. [103] empowered an edge network of various models using an LLM in a centralised cloud instance, which overlaps all the borders of the deployment space. Such a network requires a communication protocol, which impacts the system's performance when the number of devices scales up [10, 139].

Bermejo et al. [11] provide a systematic review of the use of AI on the sustainability of cloud/-fog/edge/IoT ecosystems. They mainly find a lack of consistency in the reporting and use of these models to decrease energy consumption and carbon emissions.

Zhu et al. [140] investigate the energy efficiency of Green AI for Industrial IoT. The authors propose a dynamic scheduler that distributes most resources over the edge, and they show that a well-designed scheduler can outperform the cloud with short processing speeds and low energy consumption. However, this study does not account for use cases that span more devices and the life cycle over a longer time. Mendes et al. [78] propose a similar energy-aware container scheduling algorithm for Micro-Clouds. This algorithm relieves overutilised nodes in the network in order to create a network that is overall more energy efficient.

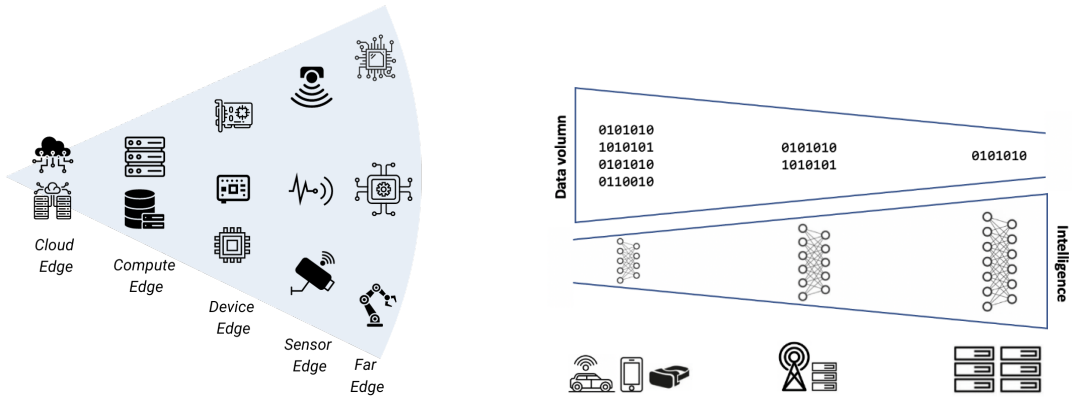
Long et al. [71] propose a complexity-aware adaptive training and inference for an edge-cloud distributed AI system. It determines the complexity of a request before sending it to the corresponding system that needs to run the inference. This adaptability decreases the energy consumption per request. Similarly, Kim et al. [60] propose the reinforcement learning algorithm AutoScale, which decides when edge inferences are executed on which device. This means inferences can be selected to run on the most energy-efficient device available, which improves energy efficiency according to the authors.

Lastly, deployment strategies such as serverless computing automatically scale nodes up and down based on usage. Patros et al. [89] highlight the need for sustainable serverless computing, as current serverless strategies can reduce their energy overhead.

Fog strategy is hard to investigate because it can vary largely in setup and it is more difficult to create an experiment which accurately reflects real-world applications [54]. This study only investigates Edge AI and Cloud AI to simplify the study and make the results interpretable.

2.4. Edge AI

Edge AI is the deployment variant that offloads the computation tasks to the respective end-user devices such that the computation can be executed close to the user and the data. This technique spans a wide array of devices visually depicted in Figure 2.2a. Devices such as phones, laptops and microcontrollers can be considered edge devices, where the user can interact directly with the models, instead of making inferences over the internet to the cloud. Figure 2.2b shows the ratio between data volume and intelligence capability, which ranges from edge devices with large amounts of data but limited resources to cloud devices where only little data is being generated compared to high-performance computational availability. In the middle Fog AI leverages the best of both worlds, however, this is out of the scope of this research. Using a combination of grey and white literature, we identified a set of advantages and challenges for Edge AI summarised in Table 2.2 included the related citations. We briefly discuss the impact of these traits on Edge AI.



(a) Types of edge devices [29].

(b) Data volume vs Intelligence capability per device type [27].

Figure 2.2: Abstractions of Edge AI from related work.

Low latency gets mentioned most often in literature and seems to be a driving factor for Edge AI, however, when we compare the current inference speeds of Cloud AI and Edge AI, they are quite different and the generally high internet speeds are not necessarily the latency bottleneck. Yet in specific situations where internet access is limited or *offline operations* are required, Edge AI is a promising technique that could increase accessibility to AI applications. Furthermore, *privacy and data sovereignty* is another important aspect of Edge AI. Due to the limited network use, the data is mostly contained on the user's device, limiting the potential of security breaches, which can be a requirement in some situations.

Another important factor, especially for the industry, is the potential *cost reduction* that can be achieved using Edge AI. We know the high cloud costs once a program scales up to a certain level, which could impact the business. Edge AI can reduce costs by offloading the work to the user device and incurring less costs on the cloud platform. However, there is no specific investigation into the actual distribution overhead of Edge AI and the cost of downloading, and maintaining deployed edge systems.

Reduced bandwidth seems like a good indication that Edge AI can impact energy consumption, as it promises reduced network traffic of the inference requests. For some AI applications, like those that deal with multimedia this could have some effect, however, for text-only applications this could

Table 2.2: Advantages and challenges of Edge AI.

Advantages	**Citations**
Low latency with users	[11, 22, 32, 35, 62, 69, 70, 76, 94, 103, 108, 112, 114, 122, 126, 140]
Privacy and data sovereignty	[22, 35, 76, 103, 112, 114, 122, 126, 128, 133, 138]
Cost efficiency	[35, 70, 76, 104, 126, 128, 133]
Offline operations (availability)	[22, 35, 104, 114, 122, 133]
Reduced internet bandwidth usage	[11, 70, 76, 94, 108]
Democratization of AI domain	[22, 138]
Personalisation	[133]
Energy efficiency	[62]
Scalability	
Challenges	
Limited computational resources edge devices	[22, 35, 36, 69, 94, 108, 112, 114, 122, 126, 127, 133, 140]
Model compression overhead	[66, 112, 114]
Amplified bias due to compression	[18, 53, 126]
Managing and updating edge devices	[122, 140]
Ensuring consistency across heterogeneous system	[35, 140]
Limited control on energy type	[128]
Scalability	

potentially be negligible. However, the overhead of downloading, and *managing* these devices and the whole *consistency* of the system could still use large amounts of internet bandwidth. Neither of these has been confirmed by research so far.

Lastly, a few small advantages include the ability to *personalise* models for better user experience, *democratise* the AI domain by distributing models, and the potential *energy efficiency* of Edge AI as they are generally less powerful devices with lower production costs compared to the cloud.

The *limited computational resources* of Edge AI pose a challenge, as not all devices have the most energy efficiency hardware for deployed AI models, which could increase energy consumption. Moreover, the Edge restricts the memory bandwidth and limits the peak computation throughput on the edge [67, 114]. Wu et al. [127] show that most phones run older and highly varied hardware, which makes the programmability harder and results in performance variability. This limitation increases the complexity of edge deployment because it can serve a wide range of customers on devices with various configurations of limited resources. This means that the developer needs to either offer a model that works on all devices set above a certain configuration or a set of differently-sized models distributed over the corresponding configuration. This could provide better performance to customers with better hardware, which poses a problem where the overall product does not produce a homogenous result.

Optimising models using quantization [67, 114] or pruning can be an effective method to enable model deployment on edge devices. These techniques are known to decrease the energy consumption for the training pipelines [115] or post-training inference but can come with an *accuracy decrease* or *bias increase* [53]. Furthermore, depending on the optimisation technique, executing the method can have an overhead of energy consumption which could impact the total energy consumption, especially if the model needs extra (re)training which is a computationally expensive task. Lastly, by moving the work of AI inference to the edge, the deployer *loses some control* over the model and how it is used. For instance, the cloud environment could use renewable energy while the edge deployment uses energy from fossil fuels, which impacts the carbon footprint more. This is a general challenge and can only realistically be solved by providing more renewable energy for everyone.

The *scalability* of Edge AI is not often mentioned in literature and lacks investigation, due to the high complexity of experimenting on them and generating a generalisable result. The resource of edge devices and the variability between devices pose real challenges for either horizontal or vertical scaling [112] and therefore need to be further investigated.

2.4.1. Green Edge AI

Due to the edge devices' inherently lower energy consumption and production costs, Edge AI is proposed as a sustainable alternative to Cloud AI. However, as we have shown there is no direct comparison between cloud and edge deployment strategies at scale. However, Edge AI has been studied for various sustainability efforts, which shows only the sustainability of Edge AI for smaller-scale projects.

Del Rey et al. [94] performed a review of the current research on the green deployment of Edge AI and found that the main limitation to Edge AI is resource restriction and there are still knowledge gaps in the field of Edge AI deployment and the factors that impact energy consumption and carbon footprint. This is acknowledged by Siemers et al. [104] who showcased the current SOTA of green mobile AI computing, showing the lack of foundational research into the sustainability of edge computing. A research agenda for trustworthy and sustainable Edge AI is proposed by Ding et al. [27], which mainly found research gaps in energy optimisations and a lack of hardware/software co-design strategies making it difficult to generalise these optimisations. Castanyer et al. [17] identified a set of design decisions that contribute to greener AI for mobile applications. The authors mainly found that increasing the complexity of the model increases resource consumption, but recommend further research with actual energy profilers.

Olliver et al. [85] explored the tradeoffs between hardware accelerators considering inference and online training on edge computing devices. They found that edge devices with GPUs are typically more energy efficient but their embodied carbon costs can outweigh the costs in scenarios where the usage is lower than average, which makes them less sustainable. Molom-Ochir et al. [80] described their experiment on the energy efficiency of various NVIDIA GPUs both edge and cloud. Their study mainly compared five different hardware accelerators to optimise for specific hardware configurations. They conclude that the GPU is more efficient than the CPU and that larger models are more efficient on larger GPUs.

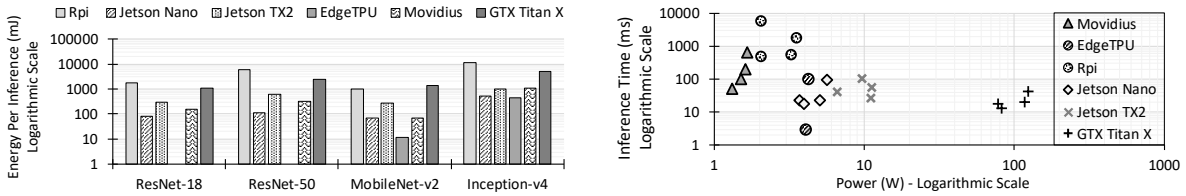
Hanafy et al. [48] found that for energy-efficient inference of Deep Neural Networks (DNN) on edge

devices, there is always a trade-off between accuracy, latency and energy. Furthermore, the authors found that this relation is non-linear, and therefore proposed an algorithm to choose the best model based on energy budget constraints. Hampau et al. [47] performed an empirical study that assesses the performance and energy consumption of various AI containerization strategies on the edge. They made recommendations for deployment strategies based on their findings. Çöplü et al. [22] tested the inference feasibility and performance of LLMs on iPhones, and although they managed to run LLMs on these devices, more work is needed to achieve an acceptable performance. Especially for battery-powered systems, AI inference can have significant impact on battery life and therefore user experience.

Gondi et al. [39] found that Automatic Speech Recognition (ASR) inference on the edge is more sustainable in terms of energy efficiency. The authors however lack evidence for this and only compare accuracy between the inference on the models, which inexplicably differs. The authors' other contribution is that small quantized models are efficient in terms of power consumption.

Hadidi et al. [46] investigated DNNs on edge deployment and the results of this study are shown in Figure 2.3, which shows the comparison of energy per inference between these devices. The authors found that between four types of edge devices and one cloud device, the cloud device uses more energy than specialised GPU-enabled edge devices, but some edge devices like the Raspberry Pi perform less energy efficiently. However, this study failed to take into account the scalability factors of AI and their potential impact on complete energy consumption.

Similarly, Lenherr et al. [64] measured the energy efficiency of Cloud and Edge AI models and found that Cloud uses 100-1000x larger power consumption for training. Furthermore, they mainly focussed on energy-precision ratios and did not compare the inference energy consumption between platforms as this would be unfair due to varying accuracies.



(a) Energy per inference for each device.

(b) Inference speed compared to power for each device.

Figure 2.3: Results of study between Cloud and Edge AI by Hadidi et al. [46].

Federated Learning

Yokoyama et al. [135] found that training ML models on an ARM-based edge device are a more cost-effective solution than on the cloud. They looked at the effects of location and user software on energy consumption, accuracy and inference time. To train larger models, a more complicated setup is required, such as Federated Learning (FL). FL is the technique of training AI models on multiple edge devices, where the edge devices each calculate portions of the weights, which are then aggregated on the Cloud or run distributed over these edge devices [86]. For smaller models, Lenherr et al. [64] found that FL uses more energy than training normally on an edge device. However, comparing it to training larger models on the cloud, it is still more energy efficient.

At Facebook, they found that FL is estimated to have a similar carbon footprint to training a normal big model [128]. Wang et al. [121] found a more sustainable method of cloud-edge FL methods using an auction system and quantization. However, many of the costs are incurred due to networking, which could indicate similar problems in the case of a distributed environment like Edge AI over time. Shen et al. [103] proposed a cloud-edge technique that can autonomously create and execute FL code to train new AI models. The authors showed that their model can allocate resources and perform FL effectively, but lack to show the sustainability of their solution.

These studies showed the complexity of FL setups and the potential overhead that can occur, which can impact the energy efficiency of the training process significantly. Therefore, FL is not considered a sustainable solution yet to train AI with less energy consumption. For this reason, FL remains out of scope for this research.

Devices

Because of the broad nature of IoT, many different types of specialised hardware have been created for all these applications. Similarly, smartphones and laptops have become increasingly more available and have better performance. However, this poses a challenge for the domain of Edge AI, since this means that the resource availability of these devices can vary enormously, and supporting them all out-of-the-box is a significant challenge. The hardware range of edge devices is roughly:

- between 1 and 10GB of RAM,
- ARM/x86 CPU, duo/quad-core,
- Cooling ranging from passive fans and heatsinks, to activate cooling,
- 0-4GB of (shared) VRAM

Models

Since the popularity of the transformer model architecture, LLM models have been increasingly applied in all kinds of scenarios. They are used in various contexts ranging from text to images to video. However, their inherently large sizes and computational requirements, make it difficult to deploy these kinds of models to the edge. For instance, `LLaMa2-7b` already requires 28GB GPU RAM to run in full precision, which does not fit any of the edge hardware configurations.

Optimisations

Fitting these large models on edge devices requires additional compression techniques. Järvenpää et al. [56] created a list of numerous tactics for Green AI to increase environmental sustainability as shown in Figure 2.1. Based on this and other case studies, we created a list of compression techniques that allow AI models to fit on edge devices and potentially reduce the energy consumption of these models. Table 2.3 shows this list of these compression techniques for LLMs. Additionally, system-level approaches such as parallelism and flash attention can work complementary to the optimisation methods and improve runtime efficiency [18, 108].

Quantization is the most mentioned in the reviewed literature and entails mapping the float weight values into whole integers in order to reduce storage and computational complexity [37]. *Pruning* is the technique of selecting and removing redundant parameters that have the least effect on the output. Furthermore, model *architecture* can have a significant impact on energy consumption and *knowledge distillation* is a specific kind of architecture where you train a smaller model based on a bigger model [99]. Lastly, *microarchitecture tuning* entails for instance hyperparameter optimisations but has energy efficiency as the goal instead of accuracy and *low-rank factorization* is another decomposition technique that tries to map the weights into a smaller yet effective data bundle.

From these techniques, quantization, pruning, and knowledge distillation, are deemed most prominent in increasing energy efficiency [22]. Of course, such techniques always result in a trade-off with accuracy. However, as Li et al. [66] showed, quantization techniques allow a model to be compressed without significant loss in accuracy. Moreover, Wang et al. [123] proposed a hardware-aware automated quantization technique that selects the best quantization depending on the hardware reducing latency and energy consumption, while maintaining accuracy. Because of the simplicity of the technique and the low computational effort that is required, quantization seems to provide a balanced solution to reduce the model size for edge deployment [33, 37, 62] and energy as well [28, 37, 50, 82, 136]. Furthermore, large-scale pruning or distillation is computationally expensive [18].

Name	Citations
Quantization	[18, 22, 28, 33, 37, 50, 53, 55, 56, 66, 69, 81, 82, 103, 115, 121, 123, 127, 133, 136, 138, 141]
Weight pruning	[18, 22, 37, 53, 56, 69, 103, 115, 127, 138, 141]
Architecture	[22, 37, 56, 69, 95, 103, 122, 127, 133]
Knowledge distillation	[18, 22, 37, 49, 56, 57, 69, 99, 103, 138, 141]
Microarchitecture tuning	[56, 127]
Low-Rank Factorization	[18, 141]

Table 2.3: Model compression techniques.

Architecture seems to play an important role in the energy consumption of a model, and various techniques have been proposed to search for the optimal architecture for the problem [69]. Moreover, Mixture-of-Experts (MoE) seems another viable option as an optimisation technique to reduce the memory requirement for these models [133]. However, to apply MoE you need the model to be trained as such, otherwise, the cost of retraining is high [122].

Although these techniques can reduce the model size significantly, quantization of for instance `LLaMa2-7B`, still results in a hardware requirement of 5-10GB RAM[1]. And even though there are edge devices that meet these specifications of the model, there are plenty that do not and could benefit from originally smaller models [18].

SLMs

To include a broader range of edge devices and their hardware limitations, we need to make an identification of available SLMs. Even though SLMs are a bit contradictory, there has been plenty of research and development into these models to make them more accessible to the public [36]. We made a collection of these models[2] which is open to contributions. It curates the latest SLMs describing their release, underlying architecture, optimisations, hardware requirements, and reproducibility.

A promising model found in this curation is `TinyLLaMa 1.1B`[3] which is significantly smaller than the original LLaMa model and trained on openly available data, but promises good performance and has a low training and inference cost. Moreover, it has a memory footprint of 0.5-4 GB of RAM, which means it can fit on a wide range of edge devices.

Overhead

Most of the research that investigates the sustainability of Edge AI look at specific use cases or technologies that improve energy efficiency. However, much of the research fails to acknowledge the inherent overhead that comes with Edge AI. Initially, the models are distributed to the edge devices over the internet. Furthermore, periodically the request on the edge device might need to be verified on a centralised processing unit to ensure consistency in the distributed system. Moreover, if a deviation is detected in the verification or the models require an update due to retraining, the models are redistributed over the complete network, again sending all the weights over the internet.

Fettweis et al. [31] described an increasing trend of energy consumption in ICT devices over time. This is due to the increasing usage of the devices as well as the increase in the number of devices available around the world. This underlines the importance of investigating the environmental impact of the overhead factors as these are logically also increasing with the trend. Simon et al. [106] addressed the full spectrum of software service life cycles and cover different categories of environmental impacts. This includes factors such as people and office, and hardware manufacturing and electricity usage. This shows that the overhead can have impact on the holistic process of Edge AI as well.

Olaru et al. [91] described model retraining techniques, that adapt the model to concept drift. They found that the original method of retraining the model completely is unsustainable and therefore the authors proposed an adaptation technique instead, which should remove some of the energy costs. This paper showed the potential problem of choosing an incorrect deployment strategy that can impact energy consumption.

Network

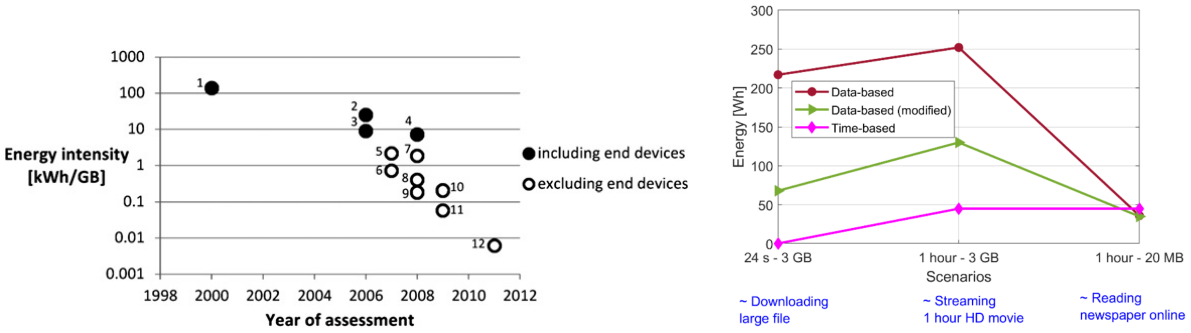
We need to investigate the energy intensity of internet traffic to enable correct calculations of the overhead of Edge AI deployment. This intensity can be expressed in either the amount of data transferred or the amount of time it takes to completely transfer the data. The latter is generally recommended for high bandwidth internet usage, like video streaming [58].

Coroama et al. [23] reviewed the methods and results of studies that assessed the energy intensity of the internet. They reviewed three methods, top-down based on estimations, model-based using simulated heuristics and lastly, and bottom-up using case studies and generalisations. The authors showed that the year of reference and the inclusion of end device energy consumption influences the results as shown in Figure 2.4a.

[1] See: `https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGUF`

[2] See: `https://rvandernoort.github.io/SmallLLMs/`

[3] See: `https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0`



(a) Energy intensity of the internet over time [23].

(b) Energy intensity of the internet for various scenarios and estimation approaches [87].

Figure 2.4: Various results of internet energy intensity.

Hinton et al. [52] proposed a model that gains insight into the contribution of different parts of the internet. The authors acknowledged the impact of data centres on energy consumption. Furthermore, they made an interesting observation that for low throughput requests, the storage disk dominates the energy consumption, while for high throughput requests the energy consumption is determined by the servers and transport network.

Oxenløwe et al. [87] explored the various methods of energy intensity estimations and proposed a standardised evaluation method for energy consumption and carbon estimation. Figure 2.4b shows the comparison of various use-case scenarios and the type of estimation technique. The authors found that the time-based methods are most useful now, but data-based models may become more important as the network becomes increasingly energy efficient.

Baliga et al. [6] provided a model-based estimation of the energy consumption of the internet by simulating the network. This study only takes limited resources into account and only provides a lower bound to the energy consumption. A later study on optical IP networks found that the energy consumption per bit of data is around 75 μJ at low access rates and decreases for higher access rates of 100 Mb/s to 2-4 μJ [7]. Lastly, the same authors analysed various types of access networks and found that the optical access network is the most energy efficient [8].

Schien et al. [100] presented a bottom-up model for the energy intensity of the Internet that draws from the current state of knowledge in the field and is specifically directed towards assessments of digital services. The authors described the total energy consumption of the network for a service as

$$E(S) = t(S) * 52W + GB(S) * 0.052kWh/GB \tag{2.1}$$

where $t(S)$ is the time of the service, $GB(S)$ the amount of data send and received by the service. The authors also showed the variable energy intensity between continuous video streaming and text retrieval, where video seems to consume more energy. Ullrich et al. [117] refined and confirmed their methods based on exemplary data and found an internet intensity of $0.02169kWh/GB$. Comparing this with previous work, they found an increase in energy efficiency due to improvements in the network and hardware. This shows that many optimisations are being deployed on the internet network which increase energy efficiency. However, currently, the network still can have significant energy usage, which could impact the energy efficiency of the deployment of a distributed system.

Measurements

We require reliable energy measurement techniques to effectively determine the energy consumption of Cloud and Edge AI. There are many options to measure the energy consumption of cloud and edge devices, but to provide a good comparison a correct strategy needs to be determined. Cabrera et al. presented the Energy Measurement Library (EML)[4] to simplify the variable measurement methods [14]. However, it lacks support for most recent hardware configurations, showing the difficulty of a single strategy across devices. Guo et al. [45] described various measurement methods for embedded devices and Chang et al. [63] described a case study for the energy measurement of an ARM processor.

[4] See: `https://github.com/HPC-ULL/eml`

Furthermore, Damaševičius et al. [25] presented an analysis of energy measurement methods for mobile devices and identified a set of challenges with current measuring techniques.

Cao et al. [15] looked into the accuracy of software-based energy measurement tools by quantifying the error using a highly accurate power meter and found that the software-based methods are not accurate and should take into account hardware variability and resource utilisation. Sallou et al. [97] created a multi-platform utility platform called `energibridge`[5], which allows energy measurements on all recent hardware configurations. This project aimed to generalise energy measurements over multiple environments, which is important for the comparability between devices. However, ARM CPUs or certain Nvidia devices still lack support.

2.4.2. Research Gap

We summarised the related work on Green AI, Cloud AI and Edge AI in which we identified a research gap. Mainly, the sustainability of Edge AI at scale in terms of energy consumption is an underinvestigated challenge. Even though Edge AI provides various advantageous properties, one of the main reasons to choose that strategy is the financial relief for the deployment costs.

However, the question remains whether Edge AI provides a good alternative to Cloud AI from an environmentally sustainable perspective. Moreover, we need to know whether it performs at scale and in which use-case scenarios it provides an energy-efficient alternative. The knowledge gap therefore lies in the absence of accurate energy consumption measurements for edge and cloud devices and a comparison between them.

[5] See: `https://github.com/tdurieux/EnergiBridge`

3

Experimental Design

To fill the identified research gap, we describe the experimental design of the research. First, we discuss the goal of this experiment, followed by the research questions that aim to fill the knowledge gap. This is then followed by the methodology, which describes the physical and software setup of the experiment, including all dependent, independent and confounding variables. Lastly, the statistical methods that are performed to compare the dataset to come to conclusions are discussed.

Cloud AI generally has a larger energy and carbon footprint than Edge AI, due to expensive hardware production costs and high base energy consumption. Generally, the cloud has a higher throughput and better parallelisation, but in practice, it suffers from underutilisation [128], which can impact the overall energy consumption. On the other hand, the high level of complexity of edge configuration with many variables such as orchestration and resource limitation is hard to simulate and measure and often does not give a complete overview of the real-world scenarios. Moreover, multiple factors determine the energy footprint of Edge AI at scale, such as model optimizations and model usage.

Based on the related work we identify four scalability factors that could determine the energy efficiency of Edge AI deployment: environment, throughput, optimisation, and model-life strategy. The environment represents the resource constraints of the device, determining the maximum model size, maximum request processing duration, and maximum throughput. For the cloud environment, this is generally abundant, while for edge devices this is limited. The throughput determines the utilisation of the model and device, which could impact the energy consumed per request as some hardware is better optimised for parallelism [128]. Optimizations for models can reduce the model size and energy consumption, and the quantization techniques provide lower inference speeds and better energy efficiency of a model [33, 62]. The model-life strategies concern initial model download and periodic verification and updating of the model. It can impact energy consumption at scale, because of the distributed nature of Edge AI [10, 128].

3.1. Methodology

The goal of this experiment and the research questions are defined using the GQM guidelines [9]. The general goal of this experiment is to: **analyse** the scalability factors of Edge AI **for the purpose of** assessing the impact **with respect to** energy efficiency **from the point of view of** AI developers **in the context of** SLM inference on the edge compared to the cloud.

This goal works towards gaining insight into (i) quantifying and evaluating the impact of the scalability factors of Edge AI with respect to energy consumption. The research objectives, variables and hypotheses are established and we perform an empirical study to answer the posed research questions. Particularly, energy consumption is measured for simulated requests to various deployed model configurations. Lastly, a simulation of the overhead factors like model download and verification is simulated. This could help us (ii) to identify the scenarios in which Edge AI and Cloud AI are most energy efficient. This could (iii) assist in a general conclusion of the energy efficiency of Edge AI at scale.

Based on this goal, the following research questions are established:

- ***RQ1:* What are the effects of architectural deployment strategies for SLMs in terms of energy consumption at scale?**

  This question aims to find determining factors for the overall energy consumption of deployment strategies by comparing Cloud AI and Edge AI. By measuring quantitative results we should be able to create an overview of the trends for multiple scalability factors that potentially impact energy consumption. The goal is to provide insight into the energy efficiency of various deployment strategies and their scalability.

  - ***RQ1.1:* How is the energy consumption impacted by the deployment environment?**
    The goal here is to find the energy difference between the cloud device and multiple edge devices running the same model. This should give us insight into the relationship between the hardware configuration and their energy efficiency for AI applications.
  - ***RQ1.2:* How is the energy consumption impacted by the quantization level?**
    The quantization levels have varying sizes and complexities and therefore energy efficiency. This question aims to find whether the quantization levels impact the energy efficiency compared to non-quantized `F32/F16` models. Furthermore, we investigate whether an optimally energy-efficient quantization level exists.
  - ***RQ1.3:* How is the energy consumption impacted by the throughput level of requests?**
    This looks at the direct scalability of requests throughput on a model. Evaluating the models under varying levels of load could show their efficiency for deployment in variable utilisation scenarios. We investigate the throughput levels from a minimum of ten requests per hour to the respective maximum throughput per device.
  - ***RQ1.4:* How is the overall energy consumption impacted when overhead factors, such as model distribution, verification, and updating are incorporated into the measurement over time?**
    This question aims to incorporate the complete model lifecycle including distribution and monitoring. We simulate the internet energy consumption that is required for various deployment strategies over the timespan of a year. The goal is to show the potential cost of using a distributed deployment strategy compared to the original cloud strategy. We achieve this by including the measurements in the simulation to find which deployment strategy is the most energy-efficient at scale.

We hypothesise that Edge AI consumes less energy for smaller projects with low throughput and low distribution over devices. However, once the project scales up this will not hold anymore and the overhead could introduce significant increases in energy. Cloud AI for high-demand applications could outperform Edge AI in energy efficiency. Although the decentralised architecture could save costs in cloud expenses, the energy usage is increased and given as responsibility to the users.

3.1.1. Variables

This section describes the independent, dependent and confounding variables of this study. They are summarised in Table 3.1 and provided with a description and scale or unit of the variables.

Independent variables

The main goal of this study is to find which devices used to deploy AI models, are the most energy efficient. Therefore, we select multiple devices ranging from High-Performance Computing (HPC) cloud devices to small microcontrollers like the Nvidia Jetson Nano (Jetson) and Raspberry Pi devices (RPi4) and (RPi5). Therefore the primary independent variable is the deployment environment, which sets the resource limitations and computational abilities.

Quantization levels determine the model size and complexity and are therefore related to the energy consumption of the models comparing non-quantized (float32/16) with quantized (>=int8). We use binary model files in GPT-Generated Unified Format (GGUF[1]) for the various quantizations as optimisation, which are available pre-quantized on the model repository HuggingFace[2]. Quantization levels

[1] See: `https://github.com/ggerganov/ggml/blob/master/docs/gguf.md`
[2] See: `https://huggingface.co/TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF`

Table 3.1: The independent, dependent, and confounding variables of this study.

Class	Name	Description	Scale
Independent	Environment	Cloud/Edge	Devices
	Quantization	GGUF	8-2 bits
	Throughput	10/max throughput	Requests/hour
	Model-life	Download, Update, and Verify over internet	Strategy
Dependent	Energy consumption	Measure/Simulate	Joules/request
Confounding	Inference framework		llama.cpp
	Dataset		OpenOrca
	Model		TinyLLama 1.1B
	EC measurement technique		AC, DC, Software
	Room temperature		Celsius

determine the model size and complexity ranging from non-quantized float 32- and 16-bit to quantized 8- to 2-bit models.

Throughput is an important factor for scalability since optimal resource utilisation improves the energy efficiency of the deployment devices. Therefore, we test various throughput levels ranging from ten requests per hour to the maximum throughput per respective device.

Lastly, the model-life strategy estimates the energy consumption for the required internet traffic. This includes model distribution, both for initial download and consequent updates of the model. Another factor is the verification of Edge models by a centralised cloud model to verify consistency over all distributed models. These variables are simulated and their consumption is estimated based on internet energy intensity estimations of previous work.

Dependent variables

We evaluate the contributions of the scalability factors of the deployment of the model concerning the energy consumption of the evaluation dataset. The energy consumptions are calculated based on the power measurements over time and are then aggregated to find the energy consumption per request. The measured variables are defined as follows:

- **GPU power**, quantifying the energy consumption of the GPU (*if applicable*).
- **CPU power**, quantifying the energy consumption of the CPU.
- **Overall energy consumption**, quantifying the total energy consumption of a device.
- **Total energy consumption**, including internet traffic energy costs.

Confounding variables

We identify that this study has confounding variables that could impact the results. The popularity of LLMs has created a large number of possible AI frameworks. These projects tackle the problem slightly differently and therefore the well-established LLM framework `llama.cpp`[3] is used for all experiments to mitigate this risk. This framework provides a low-resource solution for AI inference and therefore is suitable for Edge AI applications. It allows inference on different CUDA versions for GPU-enabled devices and offers seamless integration and quantization scripts for the `GGUF` format.

Although the dataset does not directly impact any energy consumption of the model, specific queries have seen a variable energy consumption, which could impact the results of this study. A well-established open-source dataset is chosen to ensure reproducibility. The dataset is preprocessed to remove any large queries as some overflowed the context of the models on lower resource devices[4].

There is an increased trend in SLM development due to the unavailability of LLMs on consumer-grade devices. Because of the resource limitations of edge devices, SOTA LLMs like `GPT-4` and `LLaMa2` are unable to fit. Therefore, this study opts to use a smaller implementation of the LLaMa model to

[3] See: `https://github.com/ggerganov/llama.cpp`

[4] See: `github.com/ggerganov/llama.cpp/issues/4185`

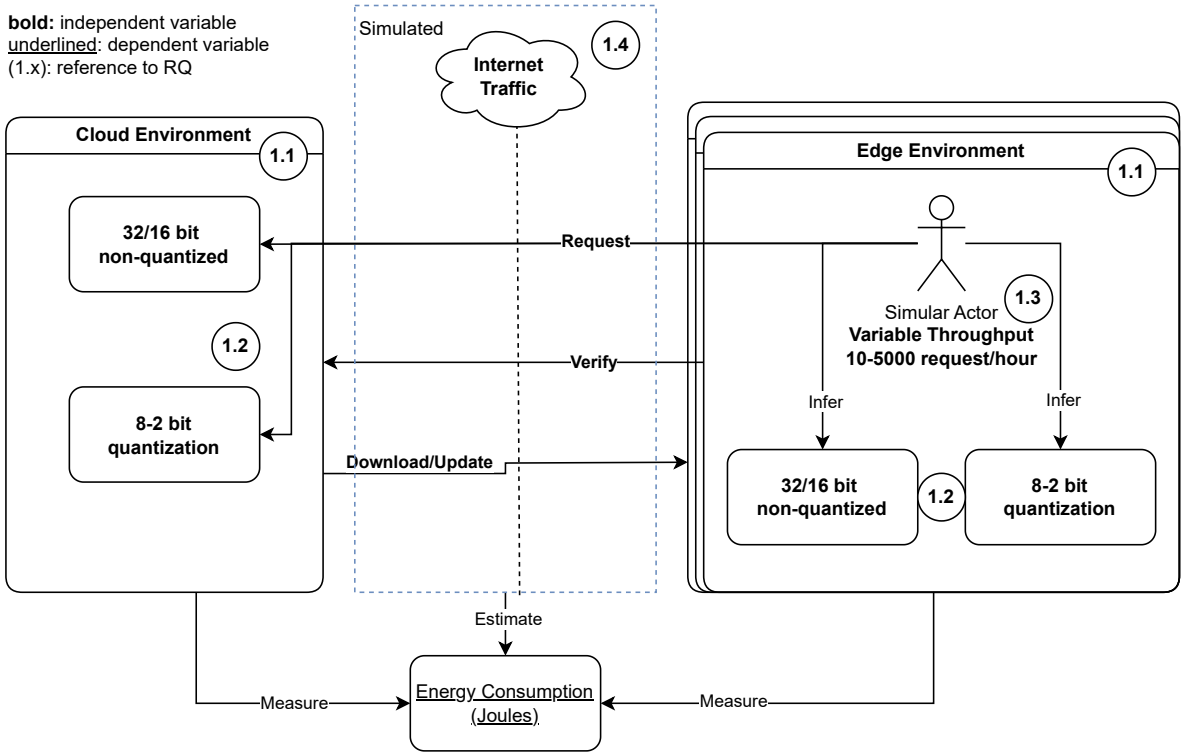


Figure 3.1: Flow chart of the experimental setup.

make fit on a resource constraint device called `TinyLLama 1.1B`[5] which is a retrained variant of `LLaMa2`, providing a small yet accurate LLM for Edge AI purposes.

Another confounding variable is the lack of a unified measurement solution for the energy consumption of all the devices. We compare the baseline consumptions of each measurement technique to ensure consistency in the results and mitigate the risk of faulty measurements.

Ambient room temperature can have a small effect on the energy consumption of hardware [90]. This confounding variable is mitigated in this study by running the experiments in a temperature-controlled room, which should alleviate any problems regarding the results.

Lastly, other variables like power, hardware utilisation, inference duration, and model size are recorded to provide more insight for the final analysis.

3.2. Experimental Setup

This study proposes the experimental setup as shown in the flowchart in Figure 3.1. The main flow consists of a single cloud environment and multiple edge devices that are directly measured on their energy consumption based on a variable request rate of inferences per hour. Different quantization levels are tested and compared to the non-quantized model's energy consumption. This allows the experiment to analyse the environment, quantization and throughput impacts on energy consumption. Lastly, we model the overhead of edge devices, such as downloading, updating, model verification and affiliated network energy consumptions in various configurations.

Firstly, the simulated actor is the starting point of the experiment and consists of a simple script that sends requests over the host network to the deployed AI model on a server on each respective device and model type. The simulated actor and the inference server are deployed using a Docker container enabling them to send messages to each other. Furthermore, the models are downloaded from Hugginface, which provides pre-quantized versions and the original model is converted to `GGUF` format using the scripts provided in `llama.cpp` into 32- and 16-bit formats. Lastly, the overhead is not implemented to measure, but this part of the study is simulated based on internet intensity measurements

[5] See: `https://github.com/jzhang38/TinyLlama`

Table 3.2: Cloud HPC and Edge devices used in the experiment.

Cloud HPC	CPU	GPU	RAM	Cooling
GreenServer	AMD Ryzen 9 7900X 12-Core	MSI RTX 4090 24G OC	64GB	Active
Edge Devices				
Raspberry Pi 4B	1.8GHz quad-core Arm Cortex-A72	None	4GB	Passive fan
Raspberry Pi 5	2.4GHz quad-core Arm Cortex-A76	VideoCore VII	8GB	Active
Nvidia Jetson Nano	quad-core Arm Cortex-A57 MPCore	Maxwell 128 CUDA cores	4GB	Passive heatsink

from previous work. These simulations and the measured energy consumption are then aggregated and used for analysis.

Hardware configurations

This experiment uses a range of devices and we summarise their hardware specifications in Table 3.2. They consist of a single Cloud HPC and multiple edge devices with a variable range of limited hardware configurations. The Cloud HPC called GreenServer has a top-of-the-line hardware configuration which should be able to handle larger models with ease. This should emulate the Cloud environments since it has abundant hardware availability. The range of edge devices includes GPU-enabled and CPU-only devices with different memory capabilities and cooling techniques, which can all impact the energy consumption of such devices [92].

Optimally, this study would include edge devices like consumer laptops and smartphones to provide better insight into a broader range of devices that are more frequently used by an average user. However, the complexity of measuring these devices and the budget limits of this project only allow for this set of devices, which should cover a wide enough range to simulate the collective group. These selected devices allow for relatively easy measurements and are quick to set up for energy experiments. We perform a small test to find the average duration and maximum throughput per hour for each device so we can test these devices at various throughput interval levels.

Models

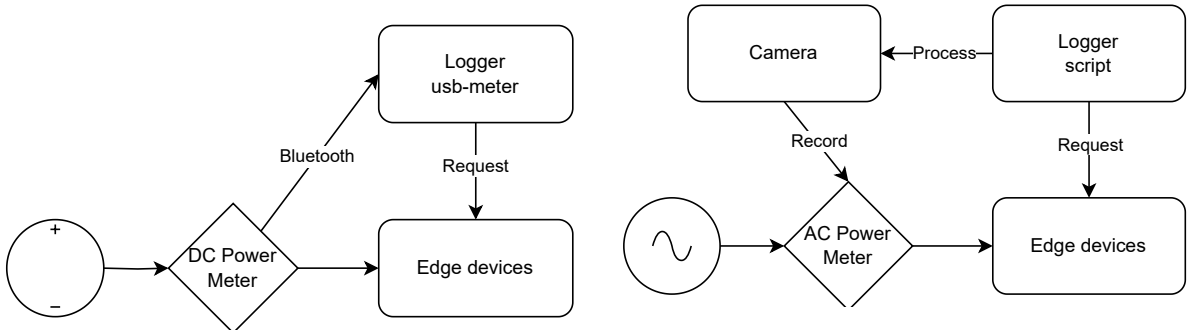
The models are retrieved from Hugginface, including the original non-quantized model and the quantized versions of 8-bit and lower. For the original sized model, the conversion script of `llama.cpp` is used to convert it into 32- and 16-bit GGUF format files since these were not readily available and are required for `llama.cpp` and the comparability of the results.

Quantization algorithms map the weight values of a model from float values to integers, which decreases the storage and computational requirements. We use pre-quantized models with GGUF, which allows for the inference of various quantization levels. These quantization levels consist of type-0 and type-1, in which type-1 includes slightly more information in the weights. Furthermore, we also use K-quant[6] levels, which are predetermined variations of the level of weight compression. Some bit levels also have variations in the size indicated by **S**mall, which indicates whether the quantized model uses only the specified bit level or **L**arge or **M**edium, which uses a higher bit level for some attention layers.

Because quantization changes the precision of the weight values, it is expected that this loses some accuracy. LLM accuracy is often expressed in perplexity, which quantifies the surprise factor that a word is selected based on the context. No perplexity measurements are performed for the model of this study, and this is considered out of scope as this research focusses on Green AI. However, GGUF quantizations are compared on perplexity for other models, showing only a negligible decrease in the accuracy of these models.

For inference, we use the `llama.cpp` server application to simulate a hosted AI application in a docker container, which then gets the simulated requests from another container. To enable `llama.cpp` to work on older hardware configurations like the Nvidia Jetson Nano, issues were created and resolved to get it working on this specific hardware[7]. This however indicates the problem of the wide array of hardware configurations that Edge AI needs to support in real-world applications.

[6] See: `https://github.com/ggerganov/llama.cpp/pull/1684`
[7] See: `https://github.com/ggerganov/llama.cpp/issues/4099`



(a) External Edge measuring setup.

(b) Normalization experiment setup.

Figure 3.2: Experimental setups for measuring techniques.

Furthermore, we use the model size to perform the simulation of the internet distribution overhead that is present for Edge AI. We use data-driven internet intensity measurements from previous work and create various strategies for downloading, verifying and updating the model on the edge devices. Due to the potential complexity of architectural strategies that can be applied to these steps, we only simulate a simplified version to make a clear conclusion.

Measurements

The Cloud HPC uses the tool `energibridge`[8] which utilises the internal measuring software of AMD for the CPU and Nvidia-SMI for the GPU. These measuring techniques are reliable sources available on all newer hardware configurations and do not require external measuring hardware. This measuring technique only takes into account the CPU and GPU energy consumption, and ignores other energy consumers, like for instance motherboard I/O and fans. However, due to the hardware configurations, these consumptions are considered negligible.

For the Nvidia Jetson Nano, the tool `jetson-stats`[9] allows you to read the energy consumption of the complete board and therefore does not require external hardware as well.

Other edge devices, like the Raspberry Pi, do not have such built-in functionalities to measure their energy consumption and therefore require a more complex setup with external measurement devices. Therefore, `Atorch J7-C`[10] is used as a DC power measurement device. Figure 3.2a shows the schema of how this device is connected after a DC convertor and sends the energy data over Bluetooth which is retrieved using the open-source program `usb-meter`[11].

3.2.1. Normalization

This study assumes that the baseline energy consumption for all devices is constant and therefore does not need to be removed from the data. This provides a good insight into the holistic energy consumption of these deployment strategies by including server idle energy consumption. Because this study utilises different energy consumption methods, it is important to validate the comparability. Because the measurements are either DC- or software-based, we can use an AC measurement device to compare the other methods by placing it before the DC converters. The AC power consumption device can measure the full power range and comes with an error of a maximum of 2% [105]. However, the energy loss in DC converters and the measurement technique for the Cloud HPC that only looks at GPU and CPU power could exhibit a larger energy consumption difference between measurement techniques. Unfortunately, the measurement device is not digital and therefore requires a setup as shown in Figure 3.2b to automate the process. We take the measurements with intervals of 1 second for around 30-60 seconds when they are tasked with 10 inference requests.

The resulting measurements are aggregated and averaged for both load and idle for which we determined a threshold. The results are compared with the DC and software measurement averages to find if there are significant differences between the measurement techniques. In Table 3.3 we show the

[8] See: `https://github.com/tdurieux/EnergiBridge`
[9] See: `https://github.com/rbonghi/jetson_stats`
[10] See: `en.atorch.cn/`
[11] See: `https://github.com/rvandernoort/usb-meter`

Table 3.3: Results of comparison power consumption performance of measurement technique.

Method	Device	AC (W)		DC/Software (W)		Threshold (W)
		idle	load	idle	load	
ATorch J76	RPi4	2.5	6.6	1.99	5.80	5
ATorch J76	RPi5	3.6	10.2	2.99	9.42	5
jetson-stats	Jetson	1.6	6.9	1.68	6.50	5
energibridge	Cloud HPC	90.2	142.1	83.8	111.5	100

resulting measurements for the AC and DC/Software measurements and the corresponding threshold. The table shows comparable results between the measurement techniques, although the Cloud HPC has higher deviations, due to the generally higher consumption and the higher error rate of the AC adapter, this difference can still be considered negligible. Due to the lack of evidence that the measurement techniques are inaccurate, we can compare the results of the various measuring techniques without adjusting them.

3.2.2. Analysis strategy

We compare the experimental measurements based on the energy consumption per request. We first perform a visual check of the data distribution to see if any preprocessing is required such as outlier removal. After the preprocessing step, we perform the statistical tests as shown in Figure 3.3 in three stages. The analysis starts with the Shapiro-Wilk test to check for normality in the data distribution, which tests the null hypothesis that the distribution is from a normally distributed population. A Quantile-Quantile (QQ) plot is then used to confirm conclusions about the normality of the distribution. We check the variances of the distributions and their similarity to the normally distributed datasets. Levene's test assumes the null hypothesis that the population variances are equal between two distributions. The boxplots of the distributions are compared visually to confirm the conclusions.

We use the appropriate statistical comparison tests to find significant differences to support the answers to the research questions. In case both distributions are normally distributed and have equal variances, the comparison is done with the ANOVA, because we want to compare the means across multiple groups. If the distributions are normal but have unequal variances, we use the Welch ANOVA t-test. If the data distribution is not normally distributed, we use the non-parametric Kruskall-Wallis test to compare the distributions for statistical differences. When the results are mixed between groups we apply the most conservative method that should maintain its statistical power.

The post-hoc Dunn's test is performed to find the distributions that are more significant than the others on non-parametric distributions, and the Games-Howell test is used on parametric distributions. These are used to make conclusions about the similarity of the distributions. Lastly, we performed the post-hoc test to find the effect size using Cliff's delta, since this is a non-parametric test quantifying the amount of difference between two groups. We use the effect magnitudes of >0 - negligible, >0.147 - small, >0.33 - medium, >0.474 - large [51]. In case we observe a high effect size difference, we compare the means to get an average scalar of the Relative Mean Difference (RMD) in energy consumption.

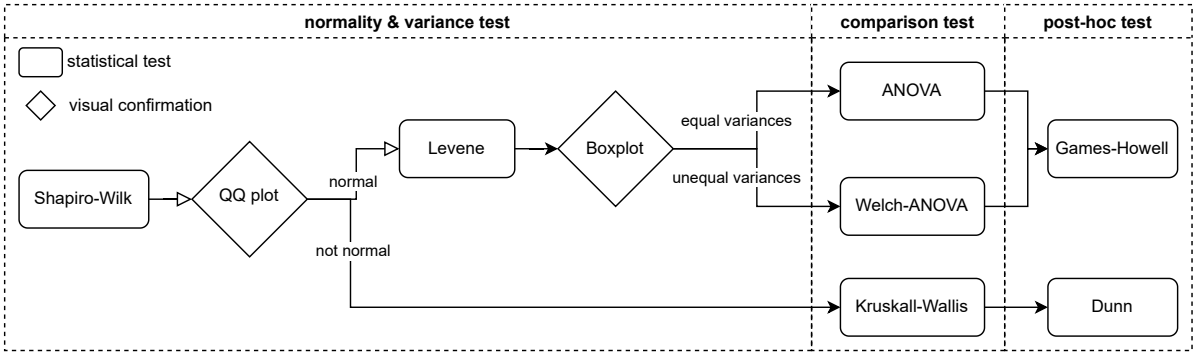


Figure 3.3: Flow chart of statistical tests

3.2.3. Replication package

To allow for the reproducibility of this study, we publish a replication package online including all the scripts to run and measure the models on the devices, preprocess the resulting data, and perform the statistical analyses. Furthermore, we include the measured and aggregated data as well in the package in case further research wants to use it. Some of the results are emitted from the analysis for simplification and clarity of the study, but they can be found in this package[12].

[12] See: https://zenodo.org/records/11065939

4

Results

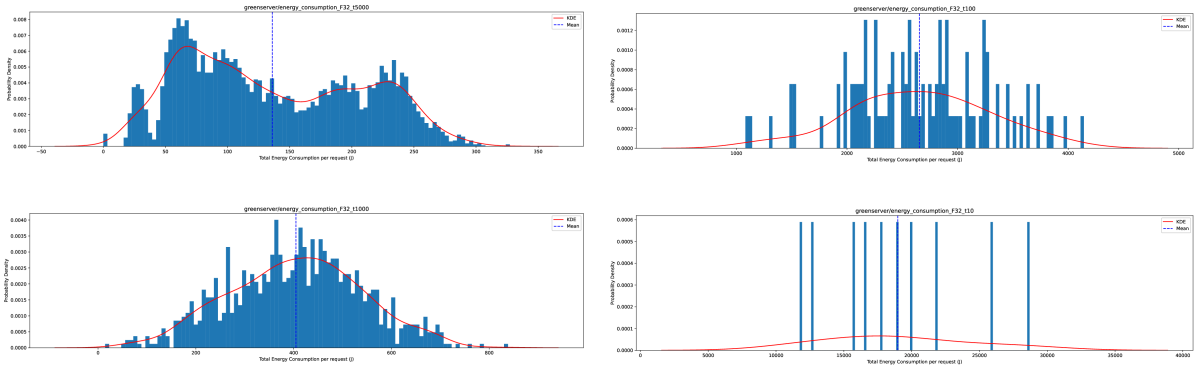
This section describes the outcomes of the experiments by analysing the resulting measurements and simulations. First, we look at the environment as the independent variable, allowing us to compare Cloud AI with Edge AI and Edge AI with each other. Then we look at the quantization level, followed by the throughput level to answer the first three research questions. Lastly, we simulate the overhead of the various deployment methods. These analyses are then combined to provide a general conclusion. For more details on the exact results of this study, Appendix A shows the index of the replication package where the statistical test results and measurement scripts are all located, which are emitted from the results for simplicity.

We performed a small throughput test on each hardware configuration and the results from that test are shown in Table 4.1. The lowest durations and highest throughputs are bold, while the highest durations and lowest throughputs are underlined. We observe that the maximum throughput of the HPC Cloud is significantly higher than any of the edge devices up to 10.000 to 100.000 requests per hour, compared to the few hundred of the edge devices.

To test the scalability of these devices, we need to include throughputs up to the maximum throughput of the device per hour. However, to accurately perform statistical tests, the experiments were limited to a maximum of 5000 requests per hour. This is because the normality and comparison statistical tests can be significantly impacted by minor deviations from normality, which makes the test results less trustworthy. However, by looking at the effect size of these distributions we should still

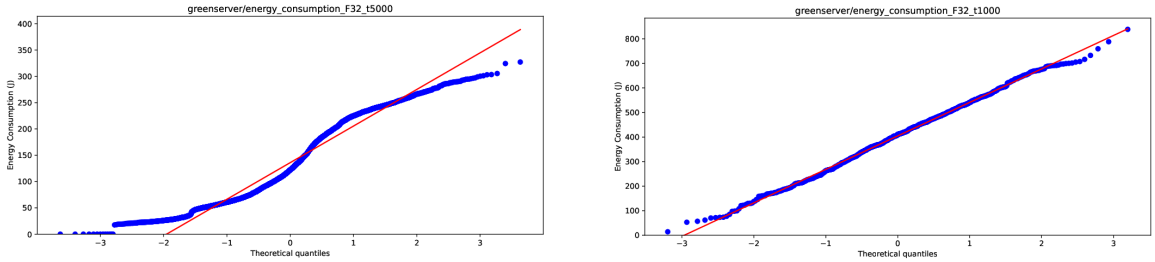
Table 4.1: Measured duration and throughputs per device and quantization level.
Dur.[1]: Average duration in ms, Thr.[2]: Average throughput in request per hour, OOM[3]: Out of Memory.

Device	**HPC Cloud**		**Jetson Nano**		**RPi5**		**RPi4**	
Quantization	**Dur.**[1]	**Thr.**[2]	**Dur.**	**Thr.**	**Dur.**	**Thr.**	**Dur.**	**Thr.**
F32	366	9836	OOM[3]		26708	247	OOM	
F16	318	11320	OOM		19513	341	OOM	
Q8_0	238	31859	23628	412	12035	224	38992	185
Q6_K	194	46652	29618	224	15906	464	40449	174
Q5_K_M	277	29243	16568	552	10824	382	35897	193
Q5_K_S	165	74945	25047	271	13103	271	36995	135
Q5_0	236	32016	25581	372	10636	581	34175	192
Q4_K_M	167	21557	24579	296	10204	390	34142	125
Q4_K_S	184	**96975**	21607	379	9840	549	30308	172
Q4_0	174	46666	**13844**	**600**	11137	441	30191	209
Q3_K_L	180	49468	21100	385	**9826**	**732**	30986	204
Q3_K_M	203	49468	22879	355	10132	372	**28741**	**223**
Q3_K_S	**152**	57423	21546	355	10166	384	31418	146
Q2_K	155	45326	17355	399	10350	385	35203	175



(a) Throughput 5000 (top) & 1000 (bot). **(b)** Throughput 100 (top) & 10 (bot).

Figure 4.1: Bell curves of energy consumption per request for HPC Cloud `F32`.



(a) Throughput 5000. **(b)** Throughput 1000.

Figure 4.2: QQ plots of HPC Cloud for `F32`.

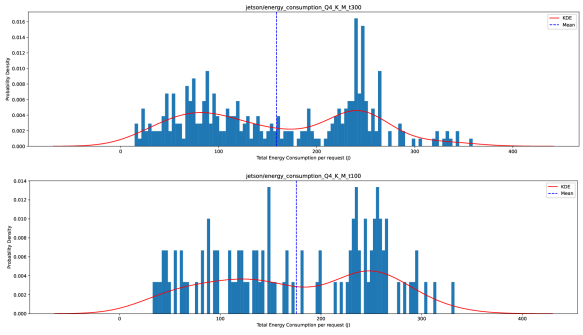
be able to determine statistical significance in scenarios with large sample sizes [111]. For comparison between edge and cloud devices, this sample size should therefore suffice as the edge devices' maximum throughput does not come close to 5000 requests per hour.

4.1. Normality

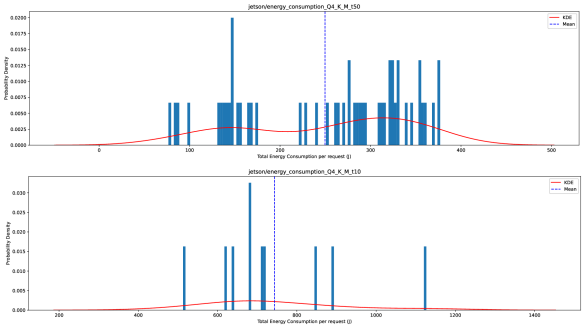
We start with the analysis of the normality of the data using the Shapiro-Wilk test for all devices, which is visually confirmed with distribution graphs and QQ plots. Firstly, the Cloud HPC is approximately normally distributed for both original and quantized models for throughputs 1000, 100 and 10 but for throughput 5000 the test shows evidence that the data does not come from a normally distributed population. As discussed, this might be due to the sample size but by performing a visual analysis of the distribution plots in Figure 4.1 we see evidence that throughput 5000 is indeed not normal, compared to the lower throughputs. In Figure 4.2 we show the QQ-plot for throughput 5000 and 1000, which shows deviation from the normal line only for throughput 5000. We emit further QQ plots from this chapter for clarity, but they can be found in the reproducibility package. Even though the quantized model shows less of a significant difference from normality, it still holds for all quantized versions.

For the Nvidia Jetson Nano (Jetson), the Shapiro-Wilk test states that most measurements seem to not be from a normally distributed population. In Figure 4.3 we show the distributions of the Jetson, and we see evidence of non-normal distributions, which is confirmed by the QQ-plots. For throughput 50, we observe some varying results between quantization levels, which shows us that low-bit quantizations are more likely to be consistent due to lower complexity.

For the Raspberry Pi 5 (RPi5) similarly, the Shapiro-Wilk tests show evidence that only throughput 10 is normally distributed. However, as shown in the distribution graphs in Figure 4.4, the distribution differs between non-quantized and quantized, from which the latter seems more likely following a normal distribution. Yet most distributions are confirmed to not be normal by the QQ plot. Only for throughput 50, we observe significance in the Shaprio-Wilk test, which shows that `F32`, `F16`, `Q3_K_L` and `Q3_K_M` are normally distributed. However, for the RPi5 we observe some outliers in the data that could contribute to this, however, by removing these from the dataset we potentially overfit the results, therefore, we deal with the distributions as is.

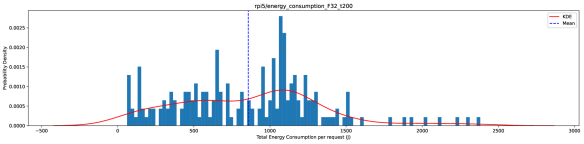


(a) Throughput 300 (top) & 100 (bot).

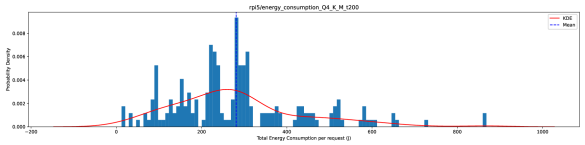


(b) Throughput 50 (top) & 10 (bot).

Figure 4.3: Bell curves of the energy consumption per request for Jetson for `Q4_K_M`.

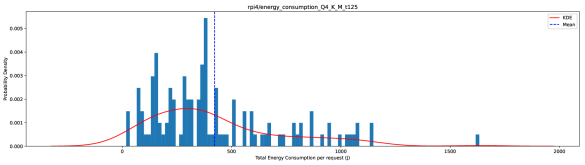


(a) Non-Quantized `F32`.

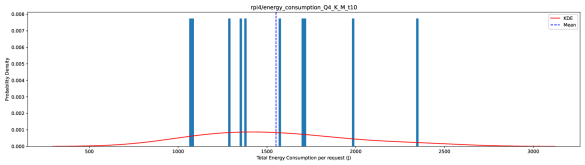


(b) Quantized `Q4_K_M`.

Figure 4.4: Bell curves of the energy consumption per request for RPi5 for throughput 200.



(a) Throughput 125.



(b) Throughput 10.

Figure 4.5: Bell curves of the energy consumption per request for RPi4 for `Q4_K_M`.

Lastly, for the Raspberry Pi 4 (RPi4), there is again evidence in the Shaprio-Wilk test that only the distribution with throughput 10 is normal. In Figure 4.5 we show the distributions for various throughputs from which we can confirm that all the other distributions for higher throughputs are not normally distributed. The increased variance compared to the other devices might be due to the limited cooling capabilities of this device, which could thermal throttle more easily than the other devices and result in outliers. In the QQ plots, we confirm that only throughput 10 is normally distributed.

4.2. Environment (RQ1.1)

First, we look at the impact of the environment on the energy consumption of a deployed model. The initial comparison is between all edge devices and the cloud, followed by a comparison between all edge devices with each other. The first should determine a difference in energy consumption between using the cloud and edge, while the next looks if a specific edge device of this set is optimal.

4.2.1. Edge vs Cloud

For all quantized models, we compare the results of all devices against each other. We emit some of the quantized results for simplicity, however, the significance is present for all quantization levels. For throughput 10, the variances between edge and cloud are compared using Levene's test, which shows significant differences between the variances. In Figure 4.6 we show the boxplot of the energy consumptions of all devices, from left to right respectively, Cloud HPC, Jetson, RPi4, and RPi5. This confirms that variances are unequal as the boxes and their ranges do not overlap.

It must be noted that we can only look at the difference between RPi5 and HPC for quantizations `F32` and `F16` because the other edge devices ran out of memory for these larger variants of the model. We already established that edge measurements were not normally distributed for throughputs above 10,

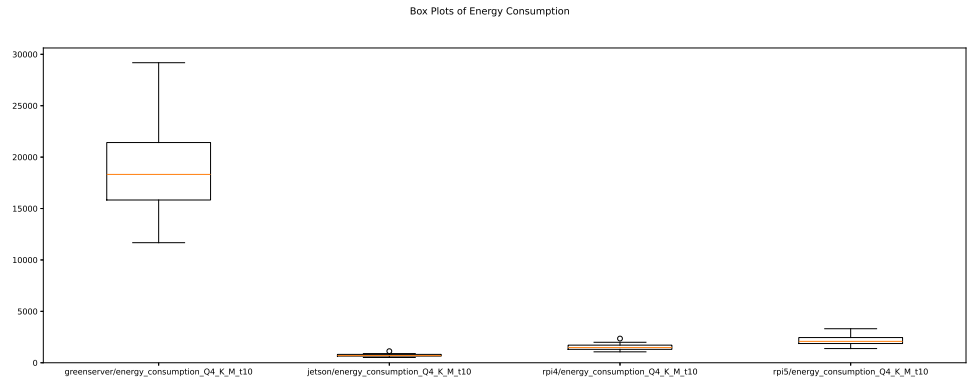


Figure 4.6: Box plot of energy consumption variances of all devices (HPC, Jetson, RPi4, RPi5) on `Q4_K_M` with throughput 10.

therefore, we used the Kruskall-Wallis test to determine similarity. Similarly to the quantized levels for the Cloud HPC and the RPi5, this unequal variance occurs for the non-quantized versions. Therefore, for all throughput 10, the Welch-ANOVA test is used for the comparison.

Table 4.2 lists the results of the relevant statistical test for each quantization level (Quant.) and throughput (Thr.), where the statistic (Stat.) or F value is the ratio of the between-group variance to the within-group variance, where a large value bigger than 1 suggests high variability. Furthermore, the p-value represents the likelihood that the distributions are drawn from the same population, where if $p < 0.05$ this hypothesis can be rejected, which is displayed in bold. Lastly, the average Relative Mean Difference (RMD) is calculated showing the effect size between cloud and edge devices.

For all non-quantized and quantized levels, there is enough evidence that shows a difference in energy consumption between the tested devices. Dunn's post-hoc test shows that the Cloud HPC is the device that impacts the results the most by having a significantly higher energy consumption than edge devices. In Figure 4.7 we plot the energy consumptions of all the devices for quantization `Q4_K_M` with throughput 100. The blue line, representing the energy consumption of the Cloud HPC, shows a large difference with the edge devices (Jetson in orange, RPi5 in green, and RPi4 in red), while these edge devices yield similar consumption patterns.

Looking at the post-hoc Cliff's delta for the effect size, we observe negligible effect size difference between RPi4 and RPi5 for a throughput higher than 10, while all other pairs have large deltas. Looking at the RMD between the edge and cloud on average we observe a 2-12x increase in energy consumption of the cloud. This means the Cloud HPC device consumes significantly more energy than the edge device for low-throughput applications. To test the same for higher throughput, the requests need to be distributed over multiple devices, multiplying the energy consumption of these devices and increasing the overhead.

Table 4.2: Statistical test results of Edge vs. Cloud.

Quan.	Thr.	Devices	Test	Stat./F	p-value	RMD
`F32`	100	RPi5/HPC	Kruskall-Wallis	136.751	**1.367e-31**	2.6x
`F32`	10	RPi5/HPC	Welch-ANOVA	90.117	**0.5e-5**	7.0x
`F16`	500	RPi5/HPC	Kruskall-Wallis	318.189	**3.592e-71**	1.7x
`F16`	100	RPi5/HPC	Kruskall-Wallis	148.955	**2.933e-34**	5.1x
`F16`	10	RPi5/HPC	Welch-ANOVA	94.971	**0.4e-5**	8.5x
`Q8_0`	100	all	Kruskall-Wallis	283.474	**3.750e-61**	9.0x
`Q4_K_M`	100	all	Kruskall-Wallis	278.973	**3.533e-60**	9.0x
`Q4_K_M`	50	all	Kruskall-Wallis	143.826	**5.656e-31**	11.6x
`Q4_K_M`	10	all	Welch-ANOVA	57.126	**4.444e-09**	8.8x
`Q2_K`	100	all	Kruskall-Wallis	275.495	**1.997e-59**	7.0x

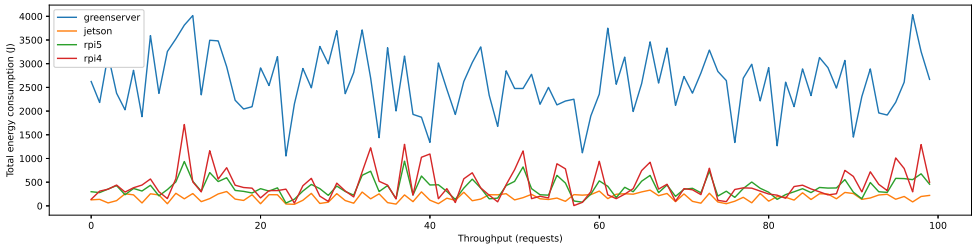


Figure 4.7: Energy consumption of all devices for `Q4_K_M` with throughput 100.

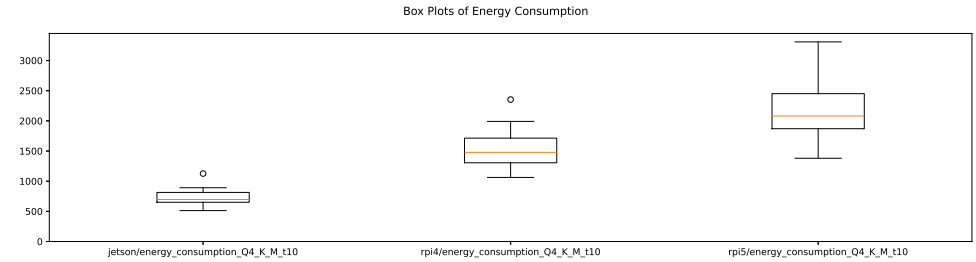


Figure 4.8: Box plot of variance of edge devices (Jetson, RPi4, RPi5 rsp.) of `Q4_K_M` with throughput 10.

4.2.2. Optimal Edge device

Next, we investigate the inter-edge devices' energy efficiency to find the effect of edge device selection. We only investigate quantized models, because of our device set only the RPi5 can run the non-quantized version. Similarly to the previous section, for throughput 10, the Welch-ANOVA test is used due to the unequal variances found in Levene's test. We verified this test by plotting the boxplot of all edge devices using the `Q4_K_M` quantization with throughput 10 in Figure 4.8. Although we can see a few similarities between the boxes from both Raspberry Pi devices, the box from Jetson confirms the hypothesis that the variances are unequal.

Table 4.3 shows the results of the statistical differences test, which show that for all quantization levels, there is a significant difference in energy consumption between edge devices. According to the respective post-hoc tests, the RPi4 and RPi5 have similar energy consumption distributions, while Jetson significantly differs from both Raspberry Pi devices. We plotted the energy consumption again for `Q4_K_M` for throughput 100 for all edge devices in Figure 4.9. We observe the same conclusions as the post-hoc test showed, where the Jetson has a consistently lower energy consumption than both Raspberry Pi devices.

Looking at the Cliff's delta post-hoc test, we observe large differences in effect size between the Raspberry Pi devices and the Jetson. By comparing the medians, we observe a decrease in energy consumption for the Jetson of about 2-3x compared to the RPi4 and RPi5. This shows that GPU-enabled devices are more energy efficient per request than CPU-only devices confirming findings from previous works.

Table 4.3: Statistical test results of comparing only the edge devices.

Quan.	Thr.	Test	Stat./F	p-value	RMD
Q8_0	100	Kruskall-Wallis	106.291	**8.301e-24**	2.1x
Q4_K_M	100	Kruskall-Wallis	97.787	**5.832e-22**	1.8x
Q4_K_M	50	Kruskall-Wallis	56.591	**5.145e-13**	1.6x
Q4_K_M	10	Welch-ANOVA	36.872	**0.2e-5**	2.1x
Q2_K	100	Kruskall-Wallis	92.681	**7.493e-21**	1.7x

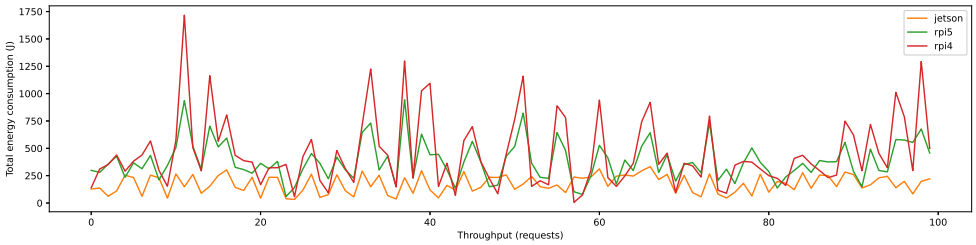


Figure 4.9: Energy consumption of edge devices for `Q4_K_M` with throughput 100.

Conclusion

We have looked at the difference in energy consumption between cloud and edge devices, where edge devices use significantly less energy than the cloud for low throughputs. Some edge devices consume even less energy than others, especially if they are GPU-enabled compared to CPU-only. Better hardware configurations for the edge seem to result in lower energy consumption for higher throughputs.

Summary 1

RQ1.1: How is the energy consumption impacted by the deployment environment?

For low throughput scenarios, edge devices consume significantly less energy for inference than the cloud. This means that applications without high throughput are recommended to use an edge device for a smaller energy footprint. The type of edge device can impact energy consumption and should be selected based on expected throughput and quantization level. This study found that CPU-only edge devices consume more energy per request than GPU-enabled edge devices.

4.3. Quantization (RQ1.2)

The next variable we look at is the quantization level and its effect on energy consumption. First, we look at the difference between non-quantized 32- and 16-bit models and all quantized versions to see if there is any indication that quantization reduces energy consumption. Next, we compare each quantization level with each other to see again if any quantization is optimal.

4.3.1. Non-quantized vs Quantized

The SLM `TinyLLama 1.1B` is significantly smaller than its base model `LLaMa 2 7B`, making it suitable for edge device deployment. However, the original unquantized versions still exceed some of these devices' memory capacities. This shows that models and device configurations are still a challenge to the broad deployment of edge devices. Moreover, more recent edge devices have less limited resource constraints and manage to run the original unquantized models. Therefore, we can compare the impact of quantization with the HPC Cloud and RPI5 devices.

Because the distribution of the Cloud HPC with throughput 5000 was not normally distributed, we used Kruskall-Wallis to test for equality. Furthermore, the HPC measurement distributions for throughput 1000 and lower are normally distributed, and therefore we perform Levene's test to compare the variances to determine the appropriate statistical test. The results of the test show that only for the non-quantized `F32` and `F16` models for throughput 1000 there is evidence that the variances are unequal compared to the quantized versions. This is confirmed by the boxplots, which are emitted for clarity, but show that HPC Cloud has unequal variances between quantization levels only for throughput 1000. This means we use Welch-ANOVA for this throughput, while we use regular ANOVA for the lower throughputs.

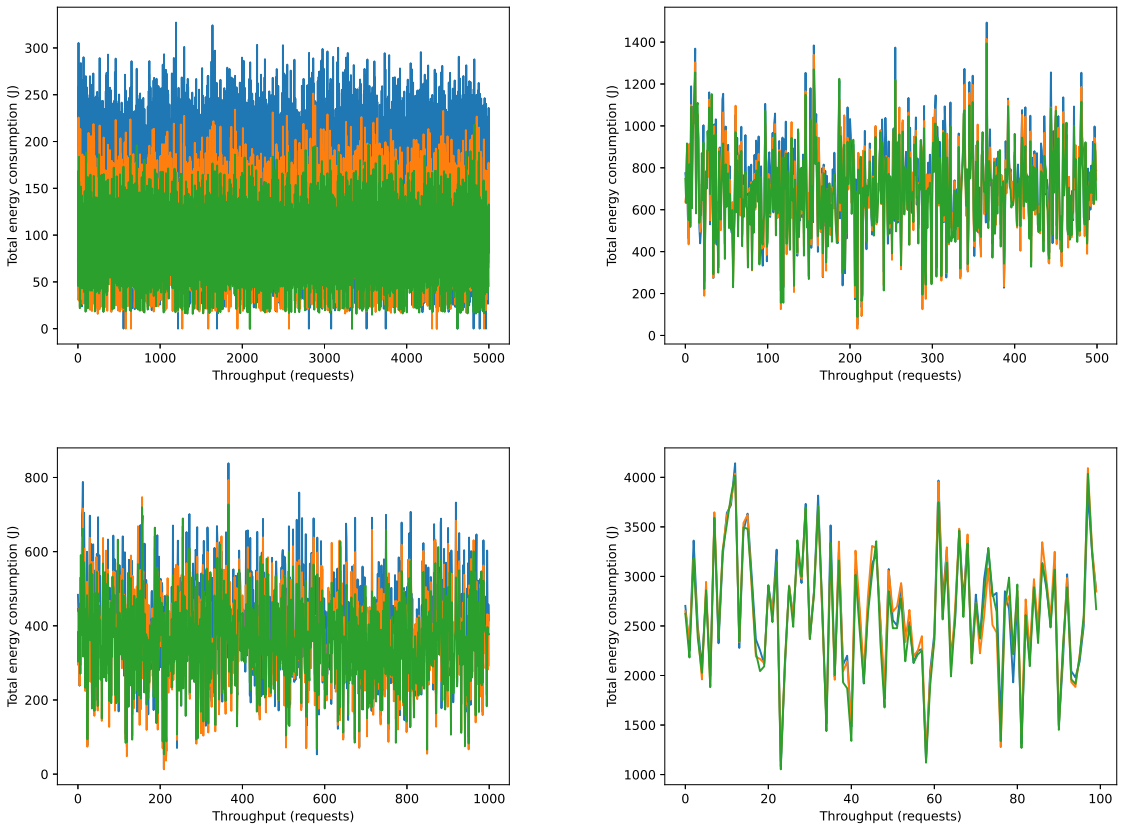
In Table 4.4 we show the results of all the respective tests to compare the non-quantized with the quantized versions. We observe that the results become significantly different only for the higher throughput levels. This could indicate that using any form of quantization on the HPC Cloud can reduce

Table 4.4: Non-Quantizated (F32,F16) vs. quantized statistical test results.

Device	Throughput	Test	Statistic/F	p-value	RMD
HPC	5000	Kruskall-Wallis	2485.523	**0.0**	1.2x
HPC	1000	Welch-ANOVA	10.008	**3.823e-21**	1.1x
HPC	500	ANOVA	5.308	**0.005**	1.0x
HPC	100	ANOVA	0.0745	0.999	1.0x
HPC	10	ANOVA	0.005	0.999	1.0x
RPI5	400	Kruskall-Wallis	805.533	**9.020e-164**	2.5x
RPI5	200	Krusakll-Wallis	339.499	**1.252e-64**	2.4x
RPI5	100	Kruskall-Wallis	150.133	**1.938e-25**	2.1x
RPI5	50	Kruskall-Wallis	69.409	**1.031e-09**	1.7x
RPI5	10	ANOVA	0.645	0.812	1.2x

energy on higher loads and does not impact energy consumption for low utilisation scenarios.

To confirm the statistical difference we found for throughput 5000, we use Dunn's post-hoc test and observe that the `F32` has the most difference with all other models, specifically with the quantized models. The 16-bit version also has a significant difference but with a lower impact. We use the Games-Howell post-hoc test for the other throughput of 1000 and 500, which finds that only the 32-bit version significantly impacts the statistical difference. In Figure 4.10 we plot the various energy consumptions of the Cloud HPC, where we show the quantization levels `F32` in blue, `F16` in orange and `Q4_K_M` in green. We can see that `F32` has a higher energy consumption for high throughputs, while we observe



(a) Throughput 5000 (top) & 1000 (bot).

(b) Throughput 500 (top) & 100 (bot).

Figure 4.10: Energy Consumptions of HPC with F32 (blue), F16 (orange), Q4KM (green) for various throughputs.

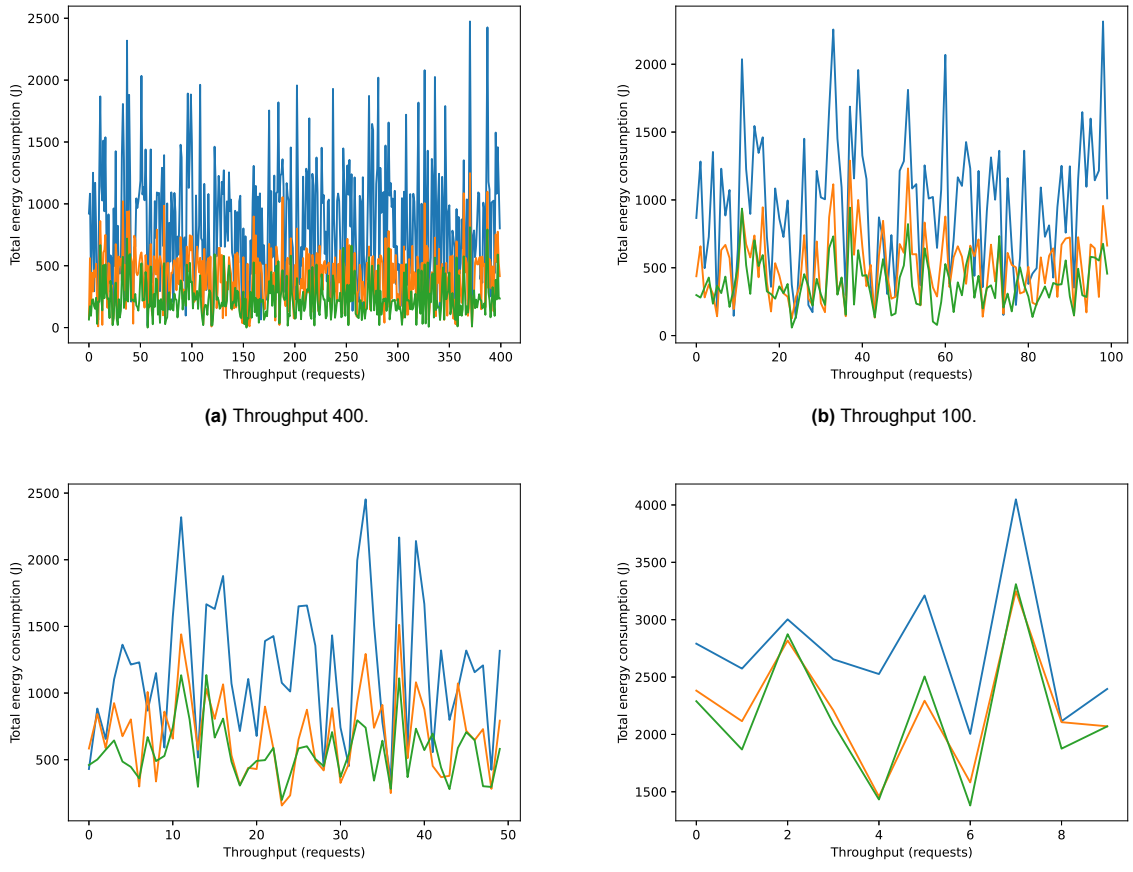
less difference for lower throughputs. This indicates that using any form of quantization on the F32 version can reduce energy consumption.

Because most throughput levels for the RPi5 are not normally distributed, we only have to test the variance of throughput 10 and both Levene's test and the box plots do not show evidence that the variance is unequal, therefore ANOVA is applied for this throughput level to test for equality.

We show in Table 4.4 that all throughputs higher than 10 have evidence of being significantly different for the RPi5. The results of the post-hoc Dunn's test confirm this by showing that F32 and F16 have the most significant difference, although for throughput 50 this significance can only be attributed to F32. This means that using quantization on the RPi5 is observed to have significant effects on the energy consumption of the model. In Figure 4.11 we plot the energy consumption of the RPi5 again for the quantization levels. We observe higher differences in energy consumption of the quantization levels for lower throughputs. Furthermore, the impact of quantization is significant for lower throughputs on the edge device compared to the Cloud HPC.

Looking at Cliff's delta for the effect size post-hoc test, we observe small and medium differences for the Cloud HPC on throughputs higher than 500, but only between the F32 level with F16 and Q4_K_M. The energy consumption of this non-quantized model is only slightly larger than the quantized version according to the RMD. For throughput 500 and lower, the effect size is considered negligible.

For the RPi5, we observe larger effect size differences and for all throughput levels. Comparing the medians, we find that quantization can reduce energy consumption by 1-3x. This shows that any quantization is more effective on edge devices, especially on higher throughputs.



(a) Throughput 400.

(b) Throughput 100.

(c) Throughput 50.

(d) Throughput 10.

Figure 4.11: Energy Consumptions of RPi5 with F32 (blue), F16 (orange), Q4_K_M (green) for various throughputs.

Table 4.5: Quantization statistical test results.

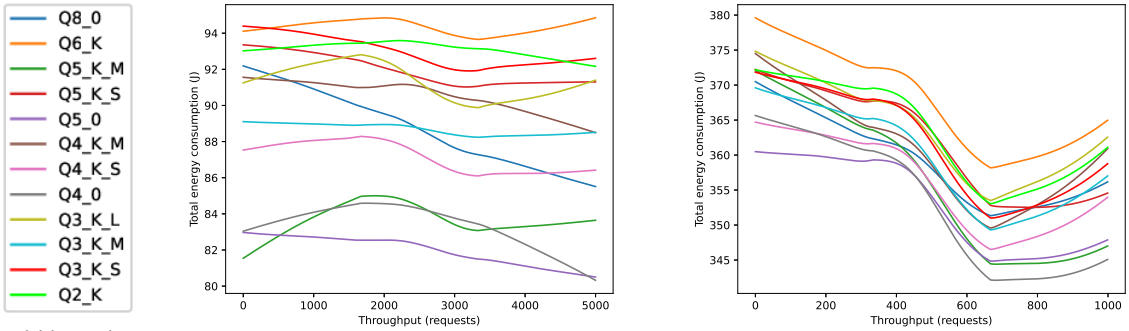
Device	Thr.	Test	Stat./F	p-value	RMD
HPC	5000	Kruskall-Wallis	715.278	**2.861e-146**	1.0
HPC	1000	ANOVA	1.417	0.157	1.0
HPC	100	ANOVA	0.067	0.999	1.0
HPC	50	ANOVA	0.006	0.999	1.0
HPC	10	ANOVA	0.005	0.999	1.0
Jetson	300	Kruskall-Wallis	70.790	**8.645e-11**	1.0
Jetson	200	Kruskall-Wallis	48.191	**1.320e-06**	1.0
Jetson	100	Kruskall-Wallis	15.869	0.146	1.0
Jetson	50	Kruskall-Wallis	20.221	**0.042**	1.0
Jetson	10	ANOVA	0.328	0.978	1.0
RPI5	400	Kruskall-Wallis	139.919	**1.692e-24**	1.1
RPI5	200	Kruskall-Wallis	49.0267	**9.358e-07**	1.0
RPI5	100	Kruskall-Wallis	20.984	**0.034**	1.1
RPI5	50	Kruskall-Wallis	8.440	0.673	1.0
RPI5	10	ANOVA	0.024	0.999	1.0
RPI4	125	Kruskall-Wallis	40.711	**2.702e-05**	1.1
RPI4	100	Kruskall-Wallis	22.648	**0.020**	1.1
RPI4	75	Kruskall-Wallis	10.863	0.455	1.0
RPI4	50	Kruskall-Wallis	9.681	0.559	1.1
RPI4	10	ANOVA	0.089	0.999	1.0

4.3.2. Optimal Quantization level

Because quantized models are smaller in model size, we can test these across a wider range of devices. We want to compare only the quantized models regarding energy efficiency to see if any quantization level is performing optimally for all the devices.

We look again at the equality of the variances between quantizations. For the Cloud HPC, we do not observe any statistically significant difference in the variances for all throughputs lower than 5000 in Levene's test and the corresponding boxplot. For all the edge devices there are no signs of significant differences between variances for throughput 10, therefore ANOVA is used for all normal distributions.

In Table 4.5 the results of each respective statistical difference test for just the quantized models on all devices are shown. We observe significant differences in all the devices for the throughputs closest to the maximum throughput per hour. For the Cloud HPC, we already observe this for throughput 5000 and in Figure 4.12 we show the Locally Weighted Scatterplot Smoothing (LOWESS) [83] graph of the respective energy runs for the quantizations. Because most distributions were not normally distributed,

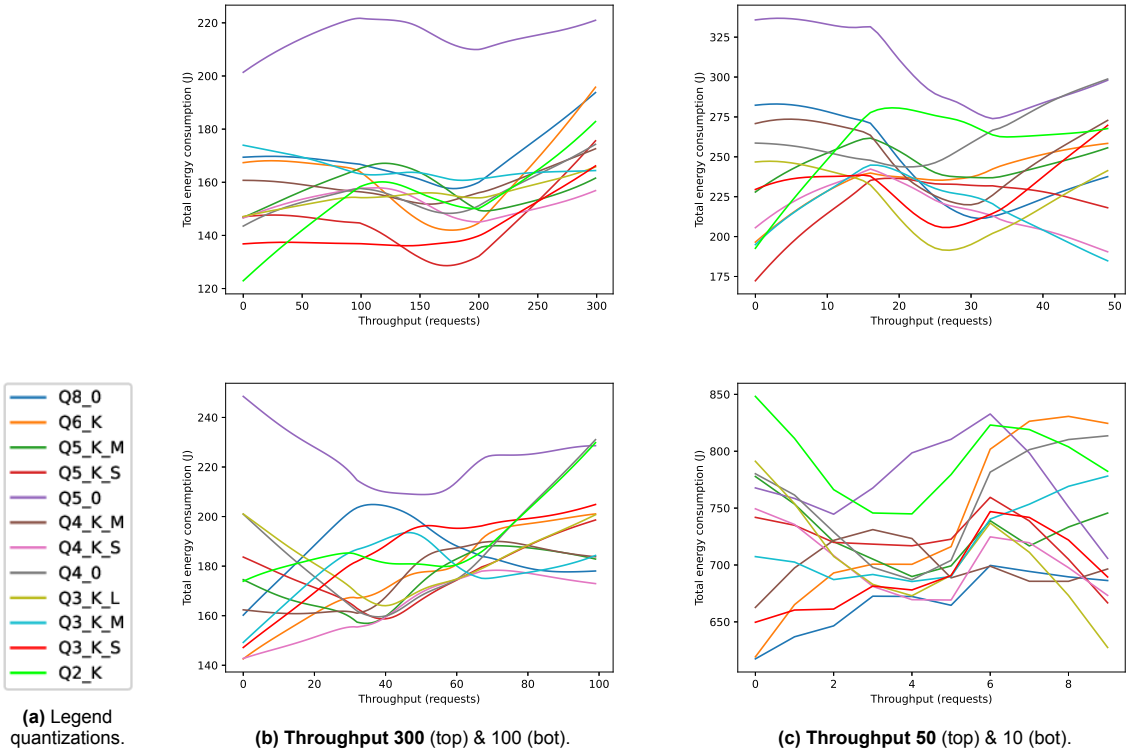


(a) Legend quantizations.

(b) Throughput 5000.

(c) Throughput 1000.

Figure 4.12: Energy Consumption per request of Cloud HPC for all quantizations.



(a) Legend quantizations.

(b) Throughput 300 (top) & 100 (bot).

(c) Throughput 50 (top) & 10 (bot).

Figure 4.13: Energy consumption Jetson for all quantizations.

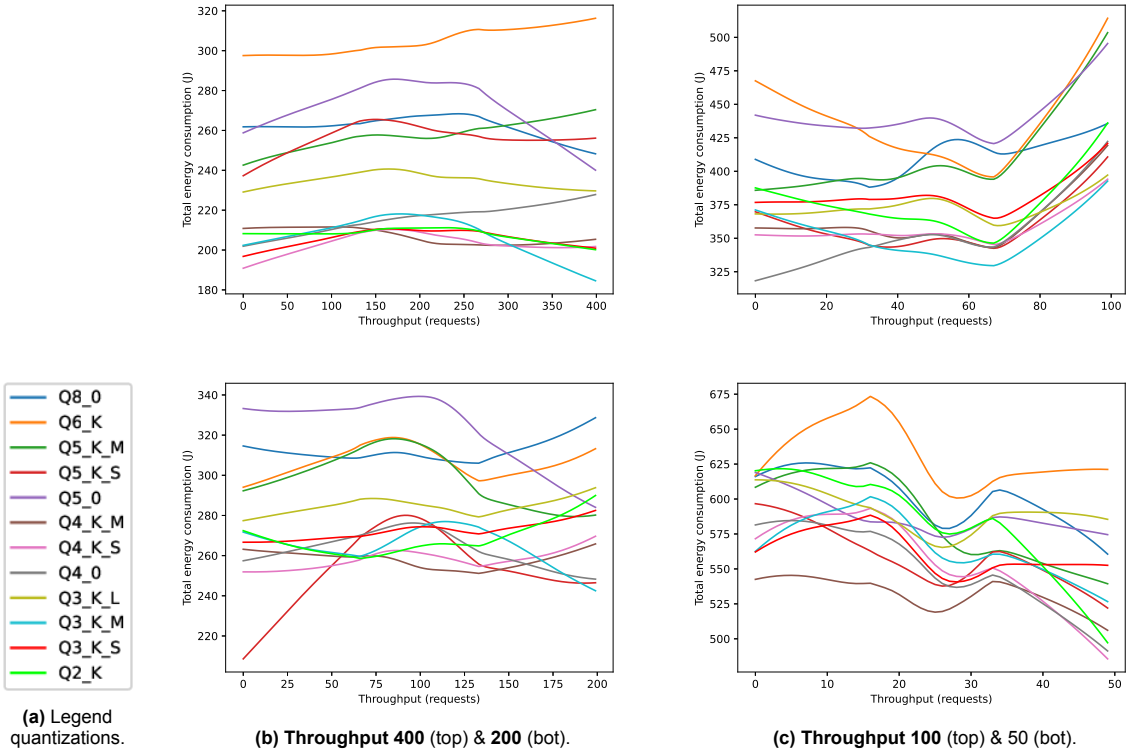
we used this smoothing function to encompass the non-parametric shape. If the figure labels are bold, the difference from the statistical test showed significance. We only show one of the insignificant results to compare and emit the other plots for clarity, but they can be found using the reproducibility package. Take into account the y-axis, which does not start at zero to be able to compare the smoothing lines. If the smoothed lines indicate higher energy consumption, and the lines are comparable then the quantization levels are less likely to significantly differ.

We observe that the energy is lower per request for higher throughputs, but they still show higher variability. This confirms the high variability on higher throughputs, while we observe no significant difference for lower throughputs. For the HPC with throughput 5000, the post-hoc using Dunn's test shows that most quantization levels are significantly different, with only a few comparable exceptions. This means that the choice of quantization level can impact the energy efficiency for higher throughputs. However, Cliff's delta post-hoc test shows only a few pairs of quantizations with a negligible effect size difference, which we additionally observe with the average RMD of 1.

If we look at the Jetson, we observe some inconsistent results. For throughput 100, the p-value does not show enough evidence to reject the hypothesis that the distributions of the quantizations are different. For all higher throughputs and throughput 50, this is the case. However, let's account for the F value, which represents the ratio between inter-difference and between-difference and should be near 1. This is for both throughput 100 and 50 relatively high, which means there is variability and evidence that the quantizations have different energy consumptions.

In Figure 4.13 we plot the energy consumptions of each quantization level, we can see the high derivation of one particular quantization level, but this does disappear with lower throughput. We see this back in the results of Dunn's test where `Q5_0` impacts the results the most, but for throughput 100 this difference is lower resulting in too low evidence to reject the null hypothesis. For throughput 300, however, there are more differences between quantization levels, but `Q5_0` is still the most prominent. Looking at Cliff's delta for this quantization level, we see it has a small impact on the difference in effect size for all throughput levels, but we find no significant deviation in the RMD as it averages to 1.

Another observation we make is that the optimal quantization level between devices differs, which could mean that the best quantization is device-dependent and cannot be generalised to select one



(a) Legend quantizations.

(b) Throughput 400 (top) & **200** (bot).

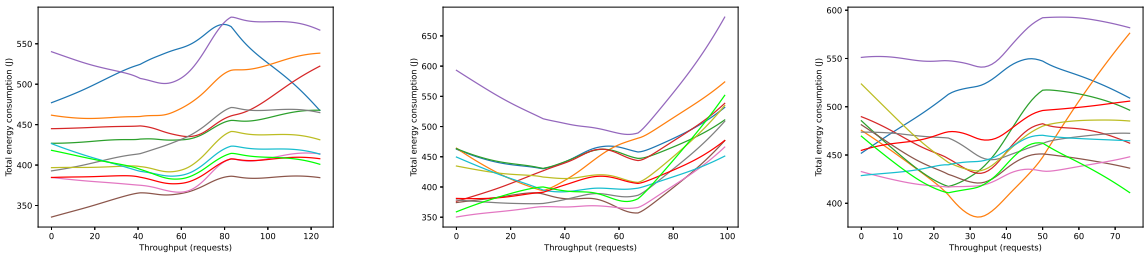
(c) Throughput 100 (top) & 50 (bot).

Figure 4.14: Energy consumption RPi5 all quantizations.

optimal level for all devices.

For the RPi5, we observe a significant difference between quantization levels from throughput 100 and higher. Figure 4.14 shows the smoothed energy consumptions per quantization level of the RPi5, which again has a higher differentiation on higher throughputs and the inconsistency of the energy efficiency of the quantization levels. Dunn's post-hoc test shows a high impact of many of the quantization algorithms, while for lower throughput this is reduced to only a few impactful levels. Cliff's delta confirms this, showing a single medium and various small differences in effect size, about 1.1x higher according to the medians for high throughput, while for lower throughputs less effect size differences occur as most are negligible. Interestingly, even the inter-device results show inconsistency in the optimal quantization level. This means it is important to account for both device and throughput when choosing a quantization level.

Similarly for the RPi4, we observe for throughputs 125 and 100, a significant difference between the energy consumption of quantizations. As shown in Figure 4.15 and in the results from Dunn's test, for the higher throughputs there is more variability between quantization levels, while this dissipates



(a) Throughput 125.

(b) Throughput 100.

(c) Throughput 75.

Figure 4.15: Energy Consumption RPi4 all quantizations.

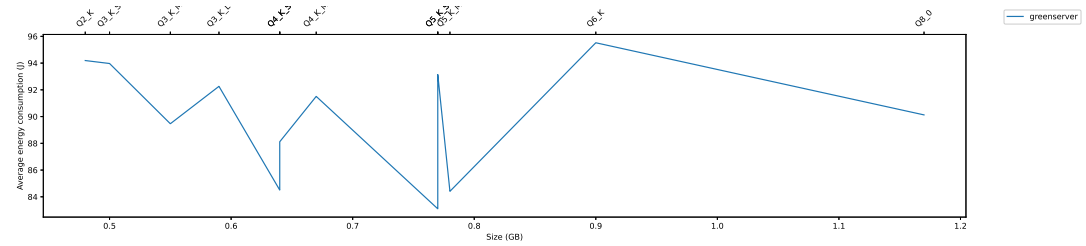
on lower throughputs. Cliff's delta shows in the post-hoc test that for all throughputs, there are pairs of quantization levels with different effect sizes, with an average RMD of 1.1.

We observe that between the edge devices, some quantization levels such as `Q8_0`, `Q6_K`, and `Q5_0` are on the higher end of the energy consumption per request, while `Q5_K_S`, `Q4_K_S`, and `Q3_K_S` are on the lower end. This is expected, since the `0` versions are based on older techniques and the `K_S` versions are small by design, logically reducing their consumption. However, for the HPC environment, some larger models seem to have a better energy efficiency, which could be due to better energy performance under higher load.

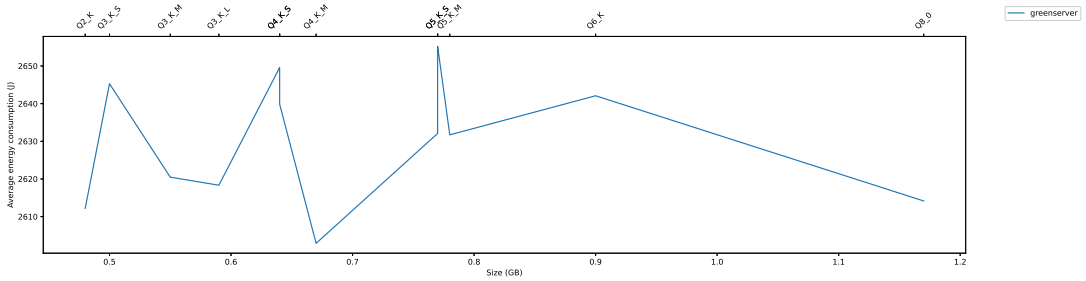
Finally, in Figure 4.16 we plot the average energy consumption per request against the model size and are labelled with the respective quantization level. We can see a variable average energy consumption per model type and size and the variability between devices and throughputs. This shows that the quantizations do not consistently or logically based on the model size, reduce energy consumption. Based on these results, the test did not show enough evidence to make any conclusions on the optimal energy-efficient quantization level.

Conclusion

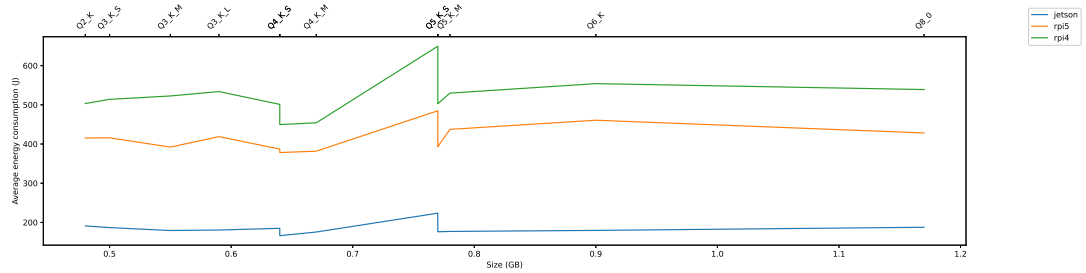
The Cloud HPC has a statistically significant lower energy consumption when using quantized models for higher load. Only on high throughput do there exist statistical differences in energy consumption between quantized models. This conclusion is similar for the edge devices, yet the throughput levels where quantization starts impacting the energy consumption are lower. Specific models perform better or worse on the edge than the cloud, but they show inconsistent results and are device- and throughput-dependent.



(a) Cloud HPC device with throughput 5000.

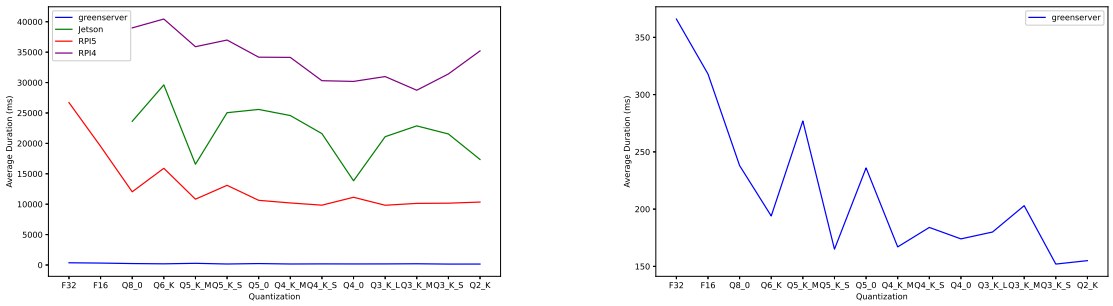


(b) Cloud HPC device with throughput 100.



(c) All edge devices with throughput 100.

Figure 4.16: Average energy consumption vs. quantized model size.



(a) Cloud HPC vs. Edge devices.

(b) Cloud HPC expanded.

Figure 4.17: Average durations per device vs. quantization level.

Summary 2

RQ1.2: How is the energy consumption impacted by the quantization level of the model?

Applying any quantization technique over standard non-quantized `F32` and `F16` bit models reduces energy consumption significantly on both cloud and edge devices for high throughputs. There is some evidence that for higher throughputs the choice of quantization level matters for all devices. There is no clear indication of the best quantization model, but this depends on the device and throughput.

4.4. Throughput (RQ1.3)

HPCs have a much higher computing capability as we've seen in the hardware specifications and this is reflected in the maximum throughput capabilities of all devices. Cloud AI can have 13-776x more throughputs per hour than Edge AI based on the throughput test we showed at the start of this chapter. The throughput for these applications significantly impacts the energy consumption of these devices, because underutilised hardware can produce idle costs that could impact the energy efficiency. Especially for HPC hardware, these costs can be quite high due to having large base power draws.

In Figure 4.17 we plot the the average duration of a request for each device and for each quantization level, which shows us significant reductions from the non-quantized `F32` and `F16` models. For the quantized levels, only a slight decrease in duration time is present for the edge devices, while this is observed to be of higher impact on the Cloud HPC, which is shown separately in Figure 4.17b.

Another observation we make is the hard resource limitation of these models, and even though the chosen model and inference framework is designed for Edge AI applications, some devices like the RPi4 and Jetson run out of memory executing the non-quantized versions of the model.

Due to the nature of the experiment throughputs are a bit harder to compare as the sample size between the datasets varies as the distributions are made by energy per request. However, we can still apply Kruskall-Wallis in this situation to compare them. We want to compare the high and low throughputs on the Cloud HPC and the edge devices to find the deployment method that fits a specific situation.

In Table 4.6 you can see that all devices show evidence that the energy consumption is affected by the actual model throughput. We can observe that an increase in the maximum throughput of a device results in a more significant difference in energy consumption. This can be confirmed with Dunn's post-hoc test, which shows that the maximum throughput and our minimum throughput of 10 per hour impact these results the most.

Cliff's delta post-hoc test shows large effect size differences between all the throughput levels. Calculating the median difference, we find that between the minimum throughput per hour and the maximum per device for the Cloud HPC, the energy per request is around 27-33x smaller on maximum throughput. The Jetson, RPi5, and RPi4 have an average RDM of around 3x. This shows that both throughput level and energy efficiency based on usage are device-dependent.

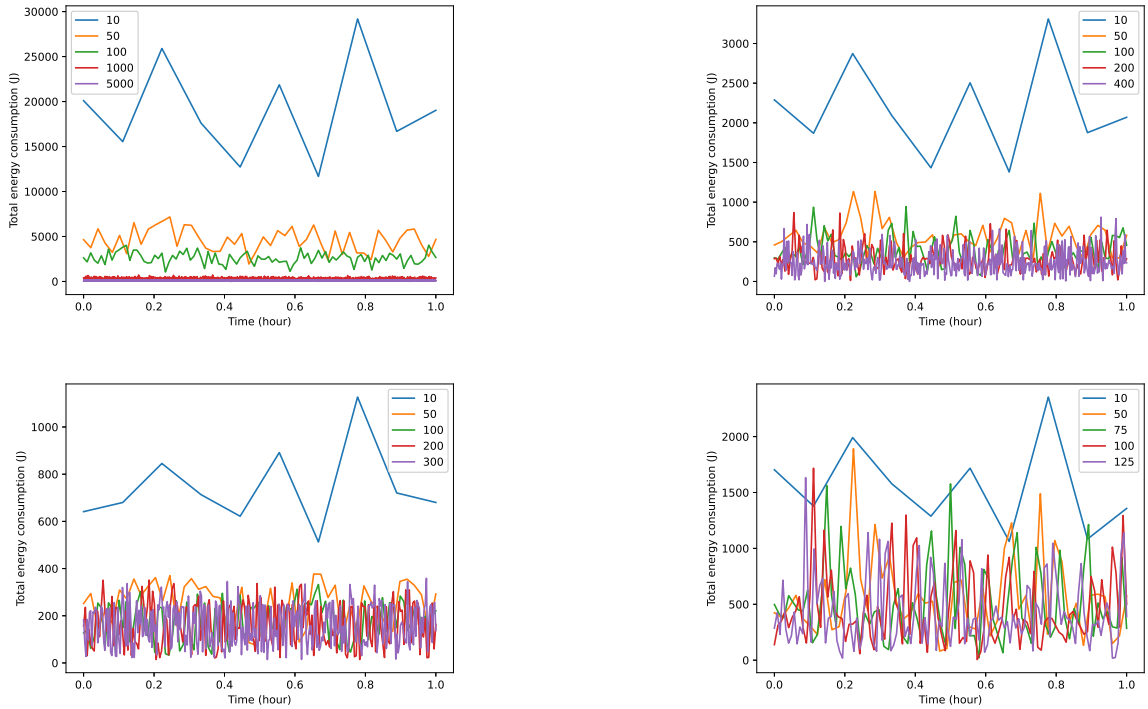
Table 4.6: Throughput statistical test results.

Device	Quant.	Test	Stat./F	p-value	RMD
HPC	F32	Kruskall-Wallis	3326.876	**0.0**	26.9x
HPC	F16	Kruskall-Wallis	3485.239	**0.0**	32.9x
HPC	Q4_K_M	Kruskall-Wallis	2721.836	**0.0**	33.9x
Jetson	Q4_K_M	Kruskall-Wallis	69.006	**3.679e-14**	2.4x
RPi5	Q4_K_M	Kruskall-Wallis	170.950	**6.541e-36**	3.7x
RPi4	Q4_K_M	kruskall-Wallis	34.782	**5.150e-07**	2.2x

We show in Figure 4.18 the energy consumption per request per device for Q4_K_M, however for all other quantization levels the same observations can be made. This graph shows the high impact of throughput especially on high-end hardware devices, while for the lower-end devices, this becomes less relevant due to their inherent limitations. Furthermore, low-throughput scenarios like throughput 10 jump out because they consume much more energy than the higher throughputs. This is because we include the idle energy cost for a holistic view of the deployment, which means low-throughput applications reduce energy consumption by being deployed on edge devices, but they still benefit from higher throughput.

Conclusion

These results indicate that the throughput of the deployed model significantly impacts the energy consumption per request. An increased throughput overall means less energy consumption per request, however, the overall consumption of the model does increase, but it can serve a high number of user requests. A reduced throughput means more energy consumption, which becomes more prevalent on high-end computing devices since their idle energy consumption is generally higher than on an edge device.



(a) HPC Cloud (top) & Nvidia Jetson Nano (bot).

(b) RaspberryPi 5 (top) and 4 (bot).

Figure 4.18: Energy Consumption for different throughputs for all devices on Q4_K_M.

Summary 3

RQ1.3: How is the energy consumption impacted by the throughput level of requests?

Higher throughput on a single model of any quantization on any device results in lower energy consumption per request. Cloud AI has 13-776x higher throughput per hour than Edge AI and therefore can serve more throughput. Some quantization levels provide higher throughput, which means they can reduce energy in case of high throughput. Low throughput scenarios use significantly more idle energy and depending on the device could impact the overall energy consumption.

4.5. Overhead (RQ1.4)

Since the Cloud HPC has significantly higher throughput, Edge AI would need multiple devices to satisfy the demand. This increases the distribution complexity.

As described in chapter 3, edge deployment comes with some overhead factors that are frequently overlooked, but should be included to create a more holistic view of the various deployment strategies. To include these overhead factors in the comparison, we simulate the energy consumption based on the latest work on internet energy intensity estimations [117].

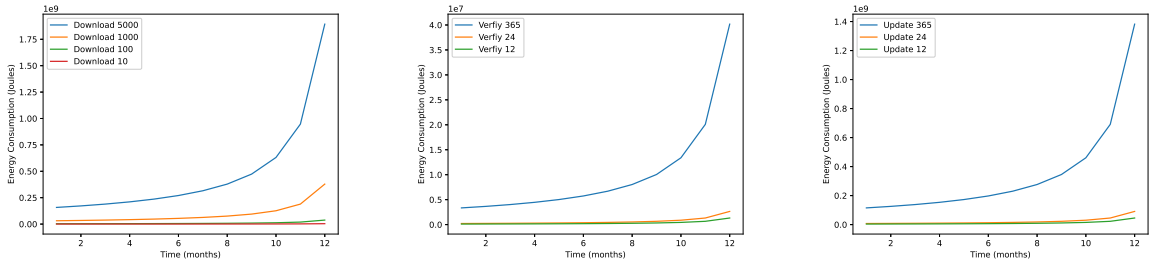
Furthermore, we model the energy consumption individually for each overhead factor, for which different strategies are applied. The resulting energy consumption simulation is shown in Figure 4.19 and is calculated based on the model size of a `Q4_K_M` of $0.67GB$, an average message size of $5KB$ and the throughput levels that we measured before. The energy consumption is simulated over the timespan of a year on the x-axis for which the amount of used devices increases linearly.

The first graph in Figure 4.19a simulates the initial download of the model for the distribution to the edge devices. We model various strategies in which a variable number of devices download the model. Logically, we see that a linear increase in devices increases energy consumption linearly as well.

Next is the periodic verification of the model, which means that the model sends a request to a Cloud instance to check for the accuracy of the distributed model. We only modelled the actual inference on a cloud device, using the average energy consumption of the `Q4_K_M` at the relevant throughput level. This means that the idle energy consumption of the measurements is not included in the simulation. The strategies vary from a verification once per month to once per day. We see in Figure 4.19b that this results in exponential energy consumption once the amount of devices scales up.

Naturally, offloading more requests from the cloud to the edge leads to lower energy consumption due to the lower consumption for edge devices, however, with only a few deployed devices, the cloud instance will have significant idle costs waiting on the verifications. When a high number of edge devices are deployed, the energy consumption per verification is expected to decrease due to optimisations. The strategy to verify can therefore have an impact on the energy consumption of the verification model.

Lastly Figure 4.19c shows the energy costs for different update strategies. Similar to the verification step it is exponential with the increasing devices and the strategy has a significant impact. However, this step consumes more energy since it periodically verifies and redownloads the model.



(a) Download simulation ranging from 10 to 5000 total downloads.

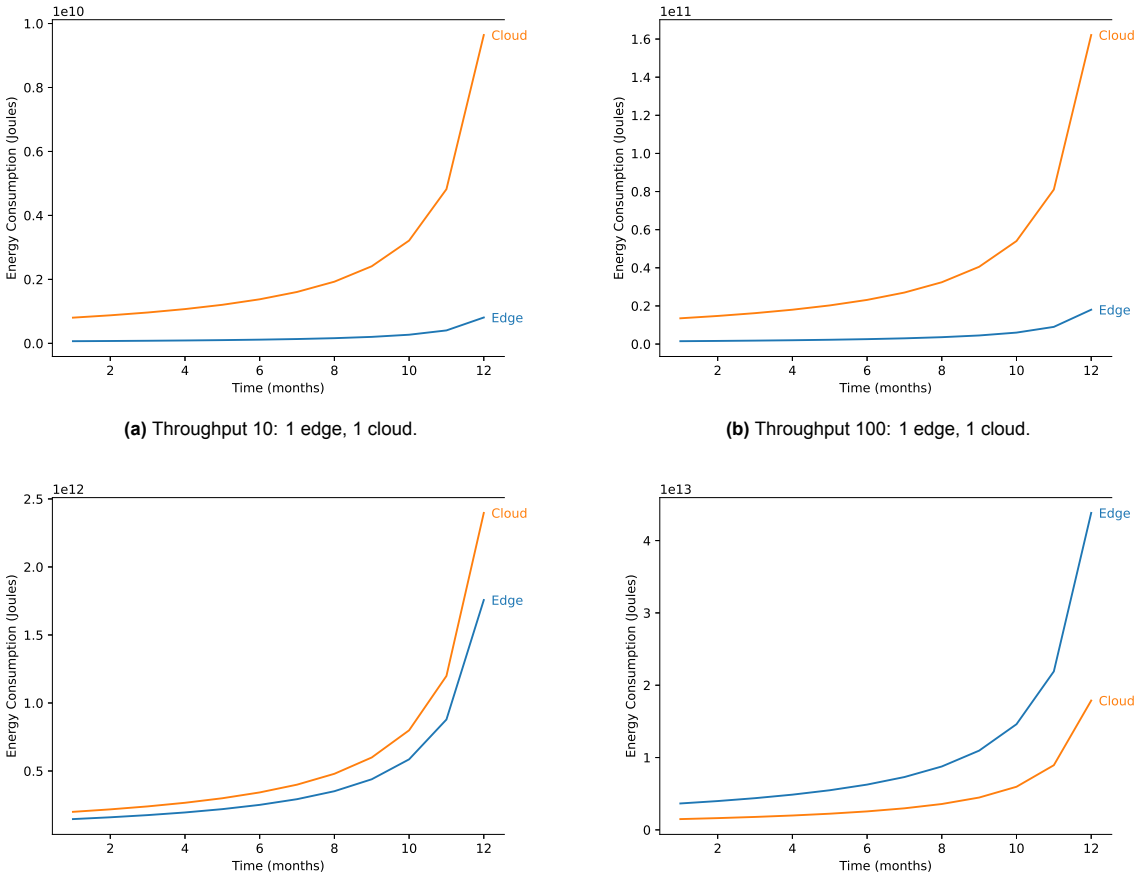
(b) Verification simulation ranging from once a month to once a day.

(c) Update simulation ranging from once a month to once a day.

Figure 4.19: Download, verify and update energy consumption cost with varying strategies.

It is possible to use a more dynamic strategy where based on the heuristics of the verification step some algorithm or model decides to perform an update, in case of drift or other similar scenarios. However, this is complex to model accurately and therefore not included in this simulation. However, it should be possible to include these scenarios in specific use cases to see the energy effect compared to these static scenarios.

Next, we investigate the impact of these overhead factors on the total energy consumption if we include them in the results from the previous measurements. In Figure 4.20 we show the combined energy consumption for `Q4_K_M` model between an edge device and a cloud device for various throughputs. Due to the limited throughput for edge devices, we modelled this by including more edge devices to achieve the wanted throughput. The figure shows that the accumulated energy consumption for the cloud is higher for low-throughput scenarios. However, at some point, the overhead of the edge devices impacts the total energy consumption and Edge AI deployment becomes less energy efficient.



(a) Throughput 10: 1 edge, 1 cloud.

(b) Throughput 100: 1 edge, 1 cloud.

(c) Throughput 1000: 10 edge, 1 cloud.

(d) Throughput 5000: 50 edge, 1 cloud.

Figure 4.20: Total energy consumption including the overhead comparison between Cloud and Edge for various throughputs.

Conclusion

This simulation has shown us the often overlooked energy impact of the overhead factors of Edge AI, such as downloading, verifying with a base model, and updating the model over time. We showed that if you include these variables with the measurements we took of these devices, there is a certain throughput level from which Edge AI becomes the less energy-efficient deployment strategy. This is due to the resource limitations and the highly optimised Cloud HPC infrastructure, and to achieve a certain level of throughput, edge devices need to be duplicated which increases this overhead significantly.

Summary 4

RQ1.4: How is the overall energy consumption impacted when overhead factors, such as model distribution, verification, and updating are incorporated into the measurement over time?

Based on simulated overhead factors, the deployment strategy can significantly impact the energy consumption of Edge AI deployment and model updating seems to have the highest impact. If we include the factors with the measurements of these devices, we see that once the deployment scales up, Edge AI is observed to use more energy than its cloud counterpart, while Edge AI consumes less for lower throughputs.

Summary 5

RQ1: What are the effects of architectural deployment strategies for SLMs in terms of energy consumption?

Based on the combination of the measurements of the experiment with different scalability factors and the simulated overhead factors, we observe that throughput and device resource restrictions are the most influential on the scalability of edge devices. It seems that Edge AI is only energy efficient in specific use cases, which are low-distribution, low-demand applications for which Edge AI significantly outperforms the cloud, but in the opposite scenario, cloud deployment is more efficient since it is better optimised for parallel performance.

5 Discussion

In the previous chapter, we showed some observations and made conclusions based on these results. In this chapter, we discuss the implications of these observations for AI developers, the threats to the study's validity, and some recommendations for future work.

5.1. Implications

In RQ1.1 we observe that edge deployment uses less energy than cloud deployment for low-throughput applications. This means Edge AI could reduce energy consumption due to its lower computing costs depending on the user scenario. However, once the scenario scales up to higher throughput and distributions the results of RQ1.4 showed that Edge AI becomes less energy efficient due to the resource limitations of edge devices and better optimizations of HPC devices.

RQ1.2 finds that applying any form of quantization reduces energy consumption. Due to the small impact on accuracy and the shown reduction in energy consumption, quantization is considered a good practice for sustainable AI deployment. More investigation is required into the optimal quantization level, which can depend on the use case. AI developers need to investigate their scenarios and find the optimal quantization level.

In RQ1.3 we observed that higher throughput reduces energy per inference for both Cloud and Edge AI deployment. This could indicate that cloud deployment could be more energy efficient for high-demand applications due to its high resource availability and higher maximum throughput than an Edge AI device. However, this means that for low-demand applications, the Cloud HPC environment uses significantly more energy than edge devices, which indicates that Edge AI is more energy efficient in these scenarios. AI developers need to account for the expected throughput and adjust their configuration accordingly.

The last observation in RQ1.4 confirms that once the demand and amount of inferences scale up, overhead factors like distribution, periodic verification, and updates impact the overall energy consumption. Our simulation shows a turning point for which deployment strategy is the most energy efficient. Edge AI is observed to consume more energy than Cloud AI in high-demand, high-distribution scenarios. This means that scalability is an effective factor in determining the sustainability of an AI deployment strategy. AI developers need to account for the overhead factors, which could add considerable energy consumption.

This results in edge deployment strategies being only environmentally sustainable in low-demand, low-throughput applications compared to cloud deployment. Developers should utilise quantization techniques if possible and find the most resource-optimal devices or optimise the model to the device for the best efficiency. For high-demand, high-throughput applications, cloud-based deployment strategies are deemed more efficient in terms of energy consumption and provide better scalability for more precise, complex and efficient models. This means that developers need to consider scalability when deciding on deployment strategies for their AI applications.

Next, we address the question of the hypothetical IT company from the introduction about which deployment strategy is most environmentally sustainable for their use cases of a coding assistant model, and their customer support chatbot. Because their code assistance application is considered a high-

demand, and high-frequency update application, edge deployment could be the less efficient variant compared to deployed in the cloud. Because the customer support chatbot lives in a low-demand, low-distribution environment, Edge AI can be considered a sustainable alternative for this application's model deployment.

5.2. Threads to validity

Internal Validity

The main limitation of this study concerns the confounding variables which narrow down the scope of this study. First, a single inference framework is used, which is actively being developed at the time of the experiment. However, this framework was one of the few that allowed for inference on all edge devices and showed good performance. Therefore, we mitigated the risk of *history* affecting the results by using a constant set of confounding variables. The model was fully released halfway through this study, which allowed us to use the final version for the most accurate measurements, which we used consistently across devices. Furthermore, the dataset used for the experiments is not specifically designed for the selected model, however, it still has the same structure and allows for a zero-shot investigation.

To avoid the *maturation* risk we performed our experiments according to the latest energy consumption measurement methods. This study aimed to standardise the measurement techniques for a fair comparison and has shown that they are similar. However, due to the external hardware, the measurements can be affected by the inaccuracies of these measurement devices.

Another limitation of this study is the constant execution time of the analysis of a single hour, which ignores the long-term effects of running hardware that can impact the overall result [92]. Due to the nature of AI, these variables could impact the results and therefore threaten this study. However, by keeping these variables constant we believe they do not significantly impact the results, which makes our results valid.

Data Validity

We preprocessed the dataset by removing larger queries since this overflowed the limited context size of the model, which is an issue within the inference framework[1]. This means this study did not investigate larger queries, which could impact the energy consumption in select use cases. It does once again show the complexity of Edge deployment for general usage.

External Validity

Furthermore, this study only looked at a single HPC device, which although it has the same level of hardware configurations as you can rent in the Cloud, still does not fully simulate the actual energy consumption in the data centres of the cloud, which could suffer from overhead or have better optimizations. However, a similar limitation exists for the edge devices because this study emulates 'all' possible edge devices with only a select range. This hardly covers all hardware configurations, however, we believe this study provides a general baseline. Furthermore, to accurately reflect real-world scenarios multiple actors should be simulated that can infer models in parallel to test the devices' capabilities of running parallel computation and its energy efficiency under this load.

Moreover, this study utilises only a single quantization technique, which makes it difficult to make a general conclusion about the effectiveness of quantization techniques or any other optimisations. During this study, newer quantization and optimisation techniques were released which could further improve the energy efficiency. Moreover, some non-quantized SLMs could not run on all edge devices due to memory overloading. This shows the difficulty of wide-scale edge deployment and the advantages of optimisation techniques.

Next, only limited throughput data points are studied in the experimentation. This means that the exact throughput level where the switching point of energy efficiency at scale occurs cannot be directly found. If we incorporate the overhead simulation, we can only conclude that for this scenario this point is between 1000 and 5000 throughputs per hour.

This study is performed in the context of SLM applications since there is a trend towards deploying these models on the edge to reduce cloud costs. However as discussed before, Edge AI provides opportunities in various contexts. This means that other model types, such as image or control models, could have different architecture and therefore another energy consumption distribution.

[1] See: `https://github.com/ggerganov/llama.cpp/issues/4185`

Construct Validity

This study does not consider the origin of the energy or the embodied carbon footprint of the devices, looking only at direct energy consumption. This makes the results comparable and interpretable because more complex metrics like CO2eq are more error-prone and harder to understand.

Additionally, edge devices on a large scale are widely available because everyone owns a smartphone or laptop, which can run some of these optimised models. This could alleviate some of the embodied carbon cost of producing these devices since they were produced for other purposes. In the future, specific hardware could be created to allow for better AI inference on these devices, which could impact carbon emissions.

However, the threat of *inaccurate operationalisation of constructs* is mitigated by having a well-established design as outlined in chapter 3 for the experiments. Furthermore, we performed a normalisation study on the measurement techniques to verify whether they produced comparable results.

Conclusion Validity

Some of the p-values in this study are significantly lower than the regular level of $\alpha = 0.05$. We already discussed some of the assumptions that could have contributed to these results, such as the high sample size, especially for the Cloud HPC. We mitigated this risk by restricting the maximum throughput for this device to 5000 requests per hour. Even though, these tests theoretically work for this throughput, however, practically the test becomes sensitive to small deviations. However, we believe the risks are mitigated by the study design and by including all the data as is, and therefore the results and conclusions are valid. Lastly, the provided replication package in subsection 3.2.3 can be used to reproduce and validate the results of this study.

5.3. Future work

These findings and limitations provide an opportunity for further research into sustainable software engineering and Green AI. Primarily, the provided replication package can be used for further study in the energy efficiency of Cloud and Edge AI. It provides the implementations of the measuring techniques, which can be utilised on the devices of this study or a broader range. Furthermore, the measurement data allows for comparisons with new data to find other insights into the sustainability of these deployment strategies. Lastly, the simulation can be used to improve the overhead estimation of Edge AI. Next, we discuss a broader collection of possible future work.

Investigate more complex Fog strategies

Almost all deployment strategies employ a hybrid system that uses a central device that aggregates, verifies, and updates edge deployments. Others deploy in the Fog on for instance access points to get the partial advantages of the edge deployment benefits. This strategy can increase the complexities of the systems and make it increasingly difficult to asses their energy consumption and carbon footprint. Therefore, more effort must be put into the research investigating the environmental sustainability of Fog deployment and other edge-cloud hybrid strategies considering their holistic energy consumption.

Broader device range

We identified the set of devices as a limitation for this study since Edge AI and Cloud AI encompass more hardware configurations than we have studied in this experiment. With the selected set we tried to cover a range as wide as possible making the results generalisable. However, because of the wide variety of edge devices used in the real world, more research should investigate other devices such as mobile devices like laptops, and phones. This is especially important for battery-powered devices since the usability of these devices is dependent on the battery life, which can be significantly impacted by Edge AI [25]. Additionally, an increasing number of devices have a dedicated GPU, which allows for better-optimised inference energy efficiency as we showed in the study. Many laptops and even phones are therefore good potential edge devices, however, this increase in computing capabilities comes with secondary carbon emission costs driven by the production of these more performant hardware configurations.

However, a wider range of edge devices increases the complexity of the study, since they possibly require their own measurement strategy, which threatens the accuracy of the comparisons of the results. Many consumer-grade laptops come with RAPL and Nvidia-SMI so that should not complicate the setup too much, however, many phones do not come with integrated measuring hardware and therefore

require an external setup like the Raspberry Pi devices. However, if we want to test the performance on battery-powered devices, like smartphones [24], the complexity significantly increases to get accurate measurements [92].

Finally, a study to investigate more complicated setups of various edge devices simultaneously dealing with various throughputs is challenging but could provide better insights into the scalability of a complete Edge AI network. This would allow us to compare the actual cloud throughputs with the aggregated edge throughputs instead of comparing simulated throughputs as done in the current research.

More optimisation techniques

Currently, the study only investigates the pre-defined quantization levels in `GGUF` from `llama.cpp`, which allows the model to fit on all tested edge devices. However, if you incorporate even more limited-resourced devices other optimisation techniques possibly need to be applied to allow models to run on these devices. Furthermore, much research is currently done in this domain and new techniques are proposed regularly, like `GPTQ`, `NF4`, and `MoE`. However, more investigation is required into the environmental sustainability of these techniques as our results showed inconsistency between the energy consumption and model size. Therefore, we recommend more investigation into the energy consumption of optimisation techniques for Edge AI.

More throughput internvals

The results showed that optimisation can affect throughput, especially on higher-end hardware. Therefore, it is important to consider the different use cases. This study investigated a selection of throughputs, whereas a more thorough investigation of more intervals can assist in finding the exact cross point on which Edge AI becomes less energy efficient than Cloud AI. However, this would require a more complicated setup with multiple edge devices being measured for a more accurate simulation of real-world applications.

Increase complexity overhead simulation

Lastly, the simulation used in this study is a basic linear model, which does not reflect real-world scenarios. Therefore, we recommend increasing the complexity of the simulations with more variables, like location, scheduling, and energy cost, and including better energy intensity estimation of the internet traffic, for instance, by modelling it like a dynamic variable.

Broader context range

Further research is required to find the generalisability of the results in other contexts since these can have different energy consumption distributions. Moreover, various kinds of contexts can differ in the overhead factors, since multimedia for instance has a significantly higher internet transportation cost than text. This shows the importance of reproducing this study for different contexts to make a general conclusion about the sustainability of Edge AI at scale.

6

Conclusion

This paper described the experiments of Edge AI deployment and the effect of scalability factors on energy consumption. We compared the environment, including an HPC Cloud, a GPU-enabled edge device, and two CPU-only edge devices. We also compared the impact of the various quantization levels and the impact of the throughput per hour on the energy consumption per request. Finally, we simulate the overhead of downloading and updating the model over time. The results show that for low-demand, low-utility scenarios Edge AI is significantly more energy efficient than Cloud AI. However, for high-demand, high-utility scenarios the Cloud AI is better optimised and requires less energy overhead than the model distribution of Edge AI. We discussed the implications of these results and finally recommended some further research.

References

[1] AI and LinkedIn community. *How can you increase edge AI sustainability?* https://www.linkedin.com/advice/0/how-can-you-increase-edge-ai-sustainability. Accessed on 2-10-2023. 2023.

[2] Mahmoud A. Albreem et al. “Green Internet of Things (GIoT): Applications, Practices, Awareness, and Challenges”. In: *IEEE Access* 9 (2021), pp. 38833–38858. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3061697.

[3] Firas Al-Ali et al. *Novel Casestudy and Benchmarking of AlexNet for Edge AI: From CPU and GPU to FPGA*. IEEE, 2020. ISBN: 9781728154428.

[4] Elarbi Badidi. *Edge AI and Blockchain for Smart Sustainable Cities: Promise and Potential*. Smart cities mostly just trafic and energy management. July 2022. DOI: 10.3390/su14137609.

[5] Jayant Baliga, Kerry Hinton, and Rodney Tucker. “Energy Consumption of the Internet”. In: July 2007, pp. 1–3. ISBN: 978-0-9775657-3-3. DOI: 10.1109/COINACOFT.2007.4519173.

[6] Jayant Baliga, Kerry Hinton, and Rodney S. Tucker. “Energy Consumption of the Internet”. In: *COIN - ACOFT 2007, Melbourne* 32 (2007).

[7] Jayant Baliga et al. “Energy consumption in optical IP networks”. In: *Journal of Lightwave Technology* 27 (13 July 2009), pp. 2391–2403. ISSN: 07338724. DOI: 10.1109/JLT.2008.2010142.

[8] Jayant Baliga et al. “Energy consumption in wired and wireless access networks”. In: *IEEE Communications Magazine* (2011).

[9] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. “The Goal Question Metric Approach”. In: 1994. URL: https://api.semanticscholar.org/CorpusID:13884048.

[10] Cüneyt Bayılmış et al. “A survey on communication protocols and performance evaluations for Internet of Things”. In: *Digital Communications and Networks* 8 (6 Dec. 2022), pp. 1094–1104. ISSN: 23528648. DOI: 10.1016/j.dcan.2022.03.013.

[11] Belen Bermejo and Carlos Juiz. *Improving cloud/edge sustainability through artificial intelligence: A systematic review*. June 2023. DOI: 10.1016/j.jpdc.2023.02.006.

[12] Brahim Betkaoui, David B. Thomas, and Wayne Luk. *Comparing Performance and Energy Efficiency of FPGAs and GPUs for High Productivity Computing*. IEEE Press, 2010. ISBN: 9781424489831.

[13] Andrew Boutros et al. “Beyond Peak Performance: Comparing the Real Performance of AI-Optimized FPGAs and GPUs”. In: Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 10–19. ISBN: 9780738105185. DOI: 10.1109/ICFPT51103.2020.00011.

[14] Alberto Cabrera et al. “Measuring energy consumption using EML (energy measurement library)”. In: *Computer Science - Research and Development* 30 (2 Apr. 2015), pp. 135–143. ISSN: 18652042. DOI: 10.1007/s00450-014-0269-5.

[15] Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. “Towards Accurate and Reliable Energy Measurement of NLP Models”. In: (Oct. 2020). URL: http://arxiv.org/abs/2010.05248.

[16] Joel Castaño et al. “Exploring the Carbon Footprint of Hugging Face’s ML Models: A Repository Mining Study”. In: (May 2023). URL: http://arxiv.org/abs/2305.11164.

[17] Roger Creus Castanyer, Silverio Martínez-Fernández, and Xavier Franch. “Which Design Decisions in AI-enabled Mobile Applications Contribute to Greener AI?” In: (Sept. 2021). URL: http://arxiv.org/abs/2109.15284.

[18] Arnav Chavan et al. “Faster and Lighter LLMs: A Survey on Current Challenges and Way Forward”. In: (Feb. 2024). URL: http://arxiv.org/abs/2402.01799.

[19] Xuntao Cheng, Bingsheng He, and Chiew Tong Lau. “Energy-efficient query processing on embedded CPU-GPU architectures”. In: Association for Computing Machinery, Inc, May 2015. ISBN: 9781450336383. DOI: 10.1145/2771937.2771939.

[20] Andrew A. Chien et al. “Reducing the Carbon Impact of Generative AI Inference (today and in 2035)”. In: Association for Computing Machinery, Inc, July 2023. ISBN: 9798400702426. DOI: 10.1145/3604930.3605705.

[21] Seth Clark. *Architecting the Edge for AI and ML*. https://medium.com/getmodzy/architecting-the-edge-for-ai-and-ml-13fccdafab96. Accessed on 2-10-2023. Mar. 2023.

[22] Tolga Çöplü et al. “A Performance Evaluation of a Quantized Large Language Model on Various Smartphones”. In: (Dec. 2023). URL: http://arxiv.org/abs/2312.12472.

[23] Vlad C. Coroama and Lorenz M. Hilty. *Assessing Internet energy intensity: A review of methods and results*. Feb. 2014. DOI: 10.1016/j.eiar.2013.12.004.

[24] Luís Miranda da Cruz and Rui Maranhão Abreu. *Tools and Techniques for Energy-Efficient Mobile Application Development*. 2019.

[25] Robertas Damaševičius, Vytautas Štuikys, and Jevgenijus Toldinas. “Methods for measurement of energy consumption in mobile devices”. In: *Metrology and Measurement Systems* 20 (3 Sept. 2013), pp. 419–430. ISSN: 08608229. DOI: 10.2478/mms-2013-0036.

[26] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. “Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning”. In: *Sustainable Computing: Informatics and Systems* 38 (Apr. 2023). ISSN: 22105379. DOI: 10.1016/j.suscom.2023.100857.

[27] Aaron Yi Ding, Marijn Janssen, and Jon Crowcroft. “Trustworthy and Sustainable Edge AI: A Research Agenda”. In: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 164–172. ISBN: 9781665416238. DOI: 10.1109/TPSISA52974.2021.00019.

[28] Ruizhou Ding et al. “Quantized Deep Neural Networks for Energy Efficient Hardware-based Inference”. In: *IEEE* (2018).

[29] Jost Elliot. *Ai at the Edge: how to bring intelligence to the edge*. https://www.modzy.com/modzy-blog/ai-at-the-edge-how-to-bring-intelligence-to-the-edge. Accessed on 19-3-2024. Aug. 2023.

[30] Daniel Escribano. *Energy consumption of machine learning deployment in cloud providers*. 2023.

[31] Gerhard Fettweis and Ernesto Zimmermann. *ICT ENERGY CONSUMPTION-TRENDS AND CHALLENGES*. 2008.

[32] Paula Fraga-Lamas, Sérgio Ivan Lopes, and Tiago M. Fernández-Caramés. “Green iot and edge AI as key technological enablers for a sustainable digital transition towards a smart circular economy: An industry 5.0 use case”. In: *Sensors* 21 (17 Sept. 2021). ISSN: 14248220. DOI: 10.3390/s21175745.

[33] Elias Frantar et al. “GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers”. In: (Oct. 2022). URL: http://arxiv.org/abs/2210.17323.

[34] Karl Freund. *GENERATIVE AI RUNNING ON EDGE AND HYBRID INFRASTRUCTURE*. https://cambrian-ai.com/wp-content/uploads/edd/2023/07/Large-Language-Models-On-Edge-Publication-FINAL.pdf. Accessed on 17-10-2023. June 2023.

[35] Karl Freund. *How To Run Large AI Models On An Edge Device*. https://www.forbes.com/sites/karlfreund/2023/07/10/how-to-run-large-ai-models-on-an-edge-device/. Accessed on 17-10-2023. June 2023.

[36] Tao Ge, Si-Qing Chen, and Furu Wei. “EdgeFormer: A Parameter-Efficient Transformer for On-Device Seq2seq Generation”. In: (Feb. 2022). URL: http://arxiv.org/abs/2202.07959.

[37] Amir Gholami et al. “A Survey of Quantization Methods for Efficient Neural Network Inference”. In: (Mar. 2021). URL: http://arxiv.org/abs/2103.13630.

[38] Paul Gillin. *Why your cloud computing costs are so high – and what you can do about them*. 2021. URL: https://siliconangle.com/2021/11/28/cloud-computing-costs-high-can/.

[39] Santosh Gondi and Vineel Pratap. "Performance and efficiency evaluation of ASR inference on the edge". In: *Sustainability (Switzerland)* 13 (22 Nov. 2021). ISSN: 20711050. DOI: 10.3390/su132212392.

[40] Google. *Carbon Footprint*. https://cloud.google.com/carbon-footprint?hl=en. Accessed on 23-10-2023. 2023.

[41] Google. *Carbon Footprint reporting methodology*. https://cloud.google.com/carbon-footprint/docs/methodology. Accessed on 24-10-2023. Oct. 2023.

[42] Google. *Carbon free energy for Google Cloud regions*. https://cloud.google.com/sustainability/region-carbon. Accessed on 23-10-2023. 2023.

[43] Google. *Reduce your Google Cloud carbon footprint*. https://cloud.google.com/architecture/reduce-carbon-footprint. Accessed on 24-10-2023. Oct. 2021.

[44] greenedge. *GreenEdge Marie Skłodowska Curie Innovative Training Network (ITN)*. https://greenedge-itn.eu. Accessed on 14-11-2023". 2020.

[45] Chen Guo et al. *A Survey of Energy Consumption Measurement in Embedded Systems*. 2021. DOI: 10.1109/ACCESS.2021.3074070.

[46] Ramyad Hadidi et al. *Characterizing the Deployment of Deep Neural Networks on Commercial Edge Devices*.

[47] Raluca Maria Hampau et al. "An empirical study on the Performance and Energy Consumption of AI Containerization Strategies for Computer-Vision Tasks on the Edge". In: Association for Computing Machinery, June 2022, pp. 50–59. ISBN: 9781450396134. DOI: 10.1145/3530019.3530025.

[48] Walid A. Hanafy, Tergel Molom-Ochir, and Rohan Shenoy. "Design Considerations for Energy-efficient Inference on Edge Devices". In: Association for Computing Machinery, Inc, June 2021, pp. 302–308. ISBN: 9781450383332. DOI: 10.1145/3447555.3465326.

[49] Sakib Haque et al. "Semantic Similarity Metrics for Evaluating Source Code Summarization". In: vol. 2022-March. IEEE Computer Society, 2022, pp. 36–47. ISBN: 9781450392983. DOI: 10.1145/nnnnnnn.nnnnnnn.

[50] Soheil Hashemi et al. "Understanding the Impact of Precision Quantization on the Accuracy and Energy of Neural Networks". In: *IEEE* (2017), pp. 1474–1479.

[51] Melinda R Hess and Jeffrey D Kromrey. *Robust Confidence Intervals 1 Robust Confidence Intervals for Effect Sizes: A Comparative Study of Cohen's d and Cliff's Delta Under Non-normality and Heterogeneous Variances*. 2004.

[52] Kerry Hinton et al. *Modeling Power Consumption of the Internet*. 2011.

[53] Sara Hooker et al. "Characterising Bias in Compressed Models". In: *CoRR* abs/2010.03058 (2020). arXiv: 2010.03058. URL: https://arxiv.org/abs/2010.03058.

[54] Hamzaoui Ikhlasse et al. *Recent implications towards sustainable and energy efficient AI and big data implementations in cloud-fog systems: A newsworthy inquiry*. Nov. 2022. DOI: 10.1016/j.jksuci.2021.11.002.

[55] Benoit Jacob et al. "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference". In: (Dec. 2017). URL: http://arxiv.org/abs/1712.05877.

[56] Heli Järvenpää et al. "A Synthesis of Green Architectural Tactics for ML-Enabled Systems". In: (Dec. 2023). URL: http://arxiv.org/abs/2312.09610.

[57] Wandri Jooste, Rejwanul Haque, and Andy Way. "Knowledge Distillation: A Method for Making Neural Machine Translation More Efficient". In: *Information (Switzerland)* 13 (2 Feb. 2022). ISSN: 20782489. DOI: 10.3390/info13020088.

[58] George Kamiya. *Factcheck: What is the carbon footprint of streaming video on Netflix?* Accessed on 14-3-2024. 2020. URL: https://www.carbonbrief.org/factcheck-what-is-the-carbon-footprint-of-streaming-video-on-netflix/.

[59] Quy Vu Khanh et al. “An efficient edge computing management mechanism for sustainable smart cities”. In: *Sustainable Computing: Informatics and Systems* 38 (Apr. 2023). ISSN: 22105379. DOI: 10.1016/j.suscom.2023.100867.

[60] Young Geun Kim and Carole Jean Wu. “Autoscale: Energy efficiency optimization for stochastic edge inference using reinforcement learning”. In: vol. 2020-October. IEEE Computer Society, Oct. 2020, pp. 1082–1096. ISBN: 9781728173832. DOI: 10.1109/MICRO50266.2020.00090.

[61] Mohit Kumar et al. *Energy-Efficient Machine Learning on the Edges*. 2020.

[62] Andrey Kuzmin et al. “Pruning vs Quantization: Which is Better?” In: (July 2023). URL: http://arxiv.org/abs/2307.02973.

[63] Naehyuck Chang Kwanho Kim Hyung Gyu Lee. *Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI £*. 2000.

[64] Nicola Lenherr, René Pawlitzek, and Bruno Michel. “New universal sustainability metrics to assess edge intelligence”. In: *Sustainable Computing: Informatics and Systems* 31 (Sept. 2021). ISSN: 22105379. DOI: 10.1016/j.suscom.2021.100580.

[65] Da Li et al. “Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs”. In: Institute of Electrical and Electronics Engineers Inc., Oct. 2016, pp. 477–484. ISBN: 9781509039364. DOI: 10.1109/BDCloud-SocialCom-SustainCom.2016.76.

[66] Shiyao Li et al. “Evaluating Quantized Large Language Models”. In: (Feb. 2024). URL: http://arxiv.org/abs/2402.18158.

[67] Ji Lin et al. “MCUNet: Tiny Deep Learning on IoT Devices”. In: (July 2020). URL: http://arxiv.org/abs/2007.10319.

[68] Weiwei Lin et al. “A cloud server energy consumption measurement system for heterogeneous cloud environments”. In: *Information Sciences* 468 (Nov. 2018), pp. 47–62. ISSN: 00200255. DOI: 10.1016/j.ins.2018.08.032.

[69] Di Liu et al. “Bringing AI to edge: From deep learning’s perspective”. In: *Neurocomputing* 485 (May 2022), pp. 297–320. ISSN: 18728286. DOI: 10.1016/j.neucom.2021.04.141.

[70] Kirsten Lloyd. *Deploy and run LLMs at the edge*. https://www.modzy.com/modzy-blog/deploy-and-run-llms-at-the-edge. Accessed on 29-09-2023. Sept. 2023.

[71] Yinghan Long et al. “Complexity-aware adaptive training and inference for edge-cloud distributed ai systems”. In: vol. 2021-July. Institute of Electrical and Electronics Engineers Inc., July 2021, pp. 573–583. ISBN: 9781665445139. DOI: 10.1109/ICDCS51616.2021.00061.

[72] Alexandra Sasha Luccioni and Alex Hernandez-Garcia. “Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning”. In: (Feb. 2023). URL: http://arxiv.org/abs/2302.08476.

[73] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. “Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model”. In: (Nov. 2022). URL: http://arxiv.org/abs/2211.02001.

[74] Kai Ma et al. “GreenGPU: A holistic approach to energy efficiency in GPU-CPU heterogeneous architectures”. In: 2012, pp. 48–57. ISBN: 9780769547961. DOI: 10.1109/ICPP.2012.31.

[75] Silverio Martínez-Fernández, Xavier Franch, and Francisco Durán. “Towards green AI-based software systems: an architecture-centric approach (GAISSA)”. In: (July 2023). URL: http://arxiv.org/abs/2307.09964.

[76] Sanjay Mazumder. *LLMOps - Generative AI needs new processes to deploy Large Language Models at the Edge*. https://www.linkedin.com/pulse/llmops-generative-ai-needs-new-processes-deploy-large-mazumder/. Accessed on 29-09-2023. May 2023.

[77] Joseph Mcdonald et al. *Great Power, Great Responsibility: Recommendations for Reducing Energy for Training Language Models*. 2022. URL: https://github.com/huggingface/trans.

[78] Seŕgio Mendes, José Simão, and Luís Veiga. “Oversubscribing Micro-Clouds with Energy-aware Containers Scheduling”. In: 19 (2019). DOI: 10.1145/3297280. URL: https://doi.org/10.1145/3297280..

[79] Sparsh Mittal and Jeffrey S. Vetter. *A survey of methods for analyzing and improving gpu energy efficiency*. Aug. 2014. DOI: 10.1145/2636342.

[80] Tergel Molom-Ochir and Rohan Shenoy. “Energy and Cost Considerations for GPU Accelerated AI Inference Workloads”. In: Institute of Electrical and Electronics Engineers Inc., 2021. ISBN: 9781665405959. DOI: 10.1109/URTC54388.2021.9701614.

[81] Bert Moons et al. *Energy-Efficient ConvNets Through Approximate Computing*. 2016.

[82] Bert Moons et al. *Minimum Energy Quantized Neural Networks*. IEEE, 2017. ISBN: 9781538618233.

[83] Gary W Moran. *Locally-Weighted-Regression Scatter-Plot Smoothing (LOWESS): A Graphical Exploratory Data Analysis Technique*. 1984.

[84] David Mytton and Masaō Ashtine. “Sources of data center energy estimates: A comprehensive review”. In: *Joule* 6.9 (2022), pp. 2032–2056. ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2022.07.011. URL: https://www.sciencedirect.com/science/article/pii/S2542435122003580.

[85] Sebastien Ollivier et al. “Sustainable AI Processing at the Edge”. In: *IEEE Micro* 43 (1 Jan. 2023), pp. 19–28. ISSN: 19374143. DOI: 10.1109/MM.2022.3220399.

[86] Safa Otoum, Ismaeel Al Ridhawi, and Hussein Mouftah. “A Federated Learning and Blockchain-Enabled Sustainable Energy Trade at the Edge: A Framework for Industry 4.0”. In: *IEEE Internet of Things Journal* 10 (4 Feb. 2023), pp. 3018–3026. ISSN: 23274662. DOI: 10.1109/JIOT.2022.3140430.

[87] Leif Katsuo Oxenløwe et al. *Evaluating Energy Consumption of Internet Services*. 2023.

[88] Łukasz Paśko et al. “Plan and Develop Advanced Knowledge and Skills for Future Industrial Employees in the Field of Artificial Intelligence, Internet of Things and Edge Computing”. In: *Sustainability (Switzerland)* 14 (6 Mar. 2022). Survey of teaching AI, IOT and EC. ISSN: 20711050. DOI: 10.3390/su14063312.

[89] Panos Patros et al. “Toward Sustainable Serverless Computing”. In: *IEEE Internet Computing* 25 (6 2021), pp. 42–50. ISSN: 19410131. DOI: 10.1109/MIC.2021.3093105.

[90] Michael K Patterson. *The Effect of Data Center Temperature on Energy Efficiency*. [IEEE], 2008. ISBN: 9781424417018.

[91] Lorena Poenaru-Olaru et al. *Retrain AI Systems Responsibly! Use Sustainable Concept Drift Adaptation Techniques*. 2023.

[92] Pijush Kanti Dutta Pramanik et al. *Power Consumption Analysis, Measurement, Management, and Issues: A State-of-the-Art Review of Smartphone Battery and Energy Usage*. 2019. DOI: 10.1109/ACCESS.2019.2958684.

[93] Murad Qasaimeh et al. “Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels”. In: *IEEE* (2019).

[94] Santiago del Rey, Silverio Martínez-Fernández, and Xavier Franch. “A review on green deployment for Edge AI”. In: *ICT4S’23: The International Conference on Information and Communications Technology for Sustainability* (2023).

[95] Santiago del Rey et al. “Do DL models and training environments have an impact on energy consumption?” In: *49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (2023).

[96] Tahereh Saheb, Mohamad Dehghani, and Tayebeh Saheb. “Artificial intelligence for sustainable energy: A contextual topic modeling and content analysis”. In: *Sustainable Computing: Informatics and Systems* 35 (Sept. 2022). ISSN: 22105379. DOI: 10.1016/j.suscom.2022.100699.

[97] June Sallou, Luís Cruz, and Thomas Durieux. “EnergiBridge: Empowering Software Sustainability through Cross-Platform Energy Measurement”. In: (Dec. 2023). URL: http://arxiv.org/abs/2312.13897.

[98] Siddharth Samsi et al. “From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference”. In: *2023 IEEE High Performance Extreme Computing Conference (HPEC)*. 2023.

[99] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: (Oct. 2019). URL: http://arxiv.org/abs/1910.01108.

[100] Daniel Schien et al. “The energy intensity of the internet: Edge and core networks”. In: *Advances in Intelligent Systems and Computing* 310 (2015), pp. 157–170. ISSN: 21945357. DOI: 10.1007/978-3-319-09228-7_9.

[101] Roy Schwartz et al. “Green AI”. In: (July 2019). URL: http://arxiv.org/abs/1907.10597.

[102] Ashutosh Sharma et al. “Sustainable smart cities: Convergence of artificial intelligence and blockchain”. In: *Sustainability (Switzerland)* 13 (23 Dec. 2021). ISSN: 20711050. DOI: 10.3390/su132313076.

[103] Yifei Shen et al. “Large Language Models Empowered Autonomous Edge AI for Connected Intelligence”. In: (July 2023). Using LLMs to understand user and execute other on edge AI systems. URL: http://arxiv.org/abs/2307.02779.

[104] Wander Siemers, June Sallou, and Luís Cruz. “The Two Faces of AI in Green Mobile Computing: A Literature Review”. In: (July 2023). URL: http://arxiv.org/abs/2308.04436.

[105] *SilverCrest Plug-in Power Meter Manuel*. URL: https://www.elektramat.nl/amfile/file/download/file/1908/product/1180885/.

[106] Thibault Simon et al. *Uncovering the Environmental Impact of Software Life Cycle*. URL: https://inria.hal.science/hal-04082263.

[107] Rajesh Singh et al. *Energy System 4.0: Digitalization of the Energy Sector with Inclination towards Sustainability*. Sept. 2022. DOI: 10.3390/s22176619.

[108] Tharmakulasingam Sirojan et al. “Sustainable Deep Learning at Grid Edge for Real-Time High Impedance Fault Detection”. In: *IEEE Transactions on Sustainable Computing* 7 (2 2022). Reduced latency higher accuracy by using smart techniques to simpify the model.., pp. 346–357. ISSN: 23773782. DOI: 10.1109/TSUSC.2018.2879960.

[109] Sami Ben Slama. *Prosumer in smart grids based on intelligent edge computing: A review on Artificial Intelligence Scheduling Techniques*. Jan. 2022. DOI: 10.1016/j.asej.2021.05.018.

[110] Mark Smith. *How edge AI is enabling cutting-edge advances in sustainability*. https://www.wevolver.com/article/how-edge-ai-is-enabling-cutting-edge-advances-in-sustainability. Accessed on 2-10-2023. Oct. 2022.

[111] Gail M. Sullivan and Richard Feinn. “Using Effect Size—or Why the P Value Is Not Enough”. In: *Journal of Graduate Medical Education* 4 (3 Sept. 2012), pp. 279–282. ISSN: 1949-8349. DOI: 10.4300/jgme-d-12-00156.1.

[112] Chellammal Surianarayanan et al. *A Survey on Optimization Techniques for Edge Artificial Intelligence (AI)*. Feb. 2023. DOI: 10.3390/s23031279.

[113] Sonal Tandon. *Environmental Reporting Dashboards for OpenStack from BBC RD*. https://superuser.openinfra.dev/articles/environmental-reporting-dashboards-for-openstack-from-bbc-rd/. Accessed on 22-10-2023. Feb. 2022.

[114] Haotian Tang et al. *TinyChat: Large Language Model on the Edge*. https://hanlab.mit.edu/blog/tinychat. Accessed on 29-09-2023. Sept. 2023.

[115] Alec Lagarde Teixidó. *ENERGY EFFICIENCY MEASUREMENT IN OPTIMIZATION AND INFERENCE OF ML MODELS*. https://upcommons.upc.edu/bitstream/handle/2117/391670/177940.pdf?sequence=2. Accessed on 22-10-2023. 2023.

[116] Shreshth Tuli et al. “HUNTER: AI based holistic resource management for sustainable cloud computing”. In: *Journal of Systems and Software* 184 (2022), p. 111124. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2021.111124. URL: https://www.sciencedirect.com/science/article/pii/S0164121221002211.

[117] Noel Ullrich et al. “Estimating the resource intensity of the Internet: A meta-model to account for cloud-based services in LCA”. In: vol. 105. Elsevier B.V., 2022, pp. 80–85. DOI: 10.1016/j.procir.2022.02.014.

[118] Jaime Vélez. *Green Ai and the Critical Role of Edge Computing in its Success*. https://barbaraiot.com/blog/green-ai-and-the-critical-role-of-edge-computing-in-its-success. Accessed on 2-10-2023. May 2023.

[119] Roberto Verdecchia, June Sallou, and Luís Cruz. "A Systematic Review of Green AI". In: (Jan. 2023). URL: http://arxiv.org/abs/2301.11047.

[120] Roberto Verdecchia et al. "Data-Centric Green AI: An Exploratory Empirical Study". In: (Apr. 2022). DOI: 10.1109/ICT4S55073.2022.00015. URL: http://arxiv.org/abs/2204.02766.

[121] Fei Wang et al. "Toward Sustainable AI: Federated Learning Demand Response in Cloud-Edge Systems via Auctions". In: Institute of Electrical and Electronics Engineers (IEEE), Aug. 2023, pp. 1–10. ISBN: 9798350334142. DOI: 10.1109/infocom53939.2023.10229014.

[122] Jiacheng Wang et al. "Toward Scalable Generative AI via Mixture of Experts in Mobile Edge Networks". In: (Feb. 2024). URL: http://arxiv.org/abs/2402.06942.

[123] Kuan Wang et al. *HAQ: Hardware-Aware Automated Quantization with Mixed Precision*. 2018.

[124] Yuxin Wang et al. "Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training". In: Institute of Electrical and Electronics Engineers Inc., May 2020, pp. 744–751. ISBN: 9781728160955. DOI: 10.1109/CCGrid49817.2020.00-15.

[125] Leonhard Wattenbach et al. "Do You Have the Energy for This Meeting? An Empirical Study on the Energy Consumption of the Google Meet and Zoom Android apps". In: May 2022.

[126] Kyle Wiggers. *The emerging types of language models and why they matter*. https://techcrunch.com/2022/04/28/the-emerging-types-of-language-models-and-why-they-matter/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAB2Cm8MY_4avnzxK7_gI00R3yBdbd0TPdDS-fn7qSWn-5bL5xhEAU0gsh22Lz_bxlQjH_5cjdReDf0-SCawSHWF2DRmdUDhHQ8b0AeBFh770xhseQgqmdMQjLrixQ6lVCoBHxGzOWNEHEtkN63wpI_Gfh0kwLSMAdgK22JtmO4Gm. Accessed on 17-10-2023. Apr. 2022.

[127] Carole Jean Wu et al. "Machine learning at facebook: Understanding inference at the edge". In: Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 331–344. ISBN: 9781728114446. DOI: 10.1109/HPCA.2019.00048.

[128] Carole-Jean Wu et al. "Sustainable AI: Environmental Implications, Challenges and Opportunities". In: (Oct. 2021). URL: http://arxiv.org/abs/2111.00364.

[129] Carole-Jean Wu et al. "Sustainable AI: Environmental Implications, Challenges and Opportunities". In: *Proceedings of Machine Learning and Systems*. Ed. by D. Marculescu, Y. Chi, and C. Wu. Vol. 4. 2022, pp. 795–813. URL: https://proceedings.mlsys.org/paper_files/paper/2022/file/462211f67c7d858f663355eff93b745e-Paper.pdf.

[130] Yinlena Xu et al. "Energy Efficiency of Training Neural Network Architectures: An Empirical Study". In: *Proceedings of the 56th Hawaii International Conference on System Sciences* (2023), pp. 781–790.

[131] Tim Yarally et al. "Batching for Green AI – An Exploratory Study on Inference". In: (July 2023). URL: http://arxiv.org/abs/2307.11434.

[132] Tim Yarally et al. "Uncovering Energy-Efficient Practices in Deep Learning Training: Preliminary Steps Towards Green AI". In: (Mar. 2023). URL: http://arxiv.org/abs/2303.13972.

[133] Rongjie Yi et al. "EdgeMoE: Fast On-Device Inference of MoE-based Large Language Models". In: (Aug. 2023). URL: http://arxiv.org/abs/2308.14352.

[134] Tan Yigitcanlar, Rashid Mehmood, and Juan M. Corchado. "Green artificial intelligence: towards an efficient, sustainable and equitable technology for smart cities and futures". In: *Sustainability (Switzerland)* 13 (16 Aug. 2021). ISSN: 20711050. DOI: 10.3390/su13168952.

[135] André M. Yokoyama et al. "Investigating hardware and software aspects in the energy consumption of machine learning: A green AI-centric analysis". In: vol. 35. John Wiley and Sons Ltd, Nov. 2023. DOI: 10.1002/cpe.7825.

[136] Ali Hadi Zadeh et al. "GOBO: Quantizing attention-based nlp models for low latency and energy efficient inference". In: vol. 2020-October. IEEE Computer Society, Oct. 2020, pp. 811–824. ISBN: 9781728173832. DOI: 10.1109/MICRO50266.2020.00071.

[137] Xiyou Zhou et al. *HULK: An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing*. 2021. URL: https://github.com/huggingface/.

[138] Zhi Zhou et al. “Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing”. In: *Proceedings of the IEEE* 107 (8 Aug. 2019), pp. 1738–1762. ISSN: 15582256. DOI: 10.1109/JPROC.2019.2918951.

[139] Ruijie Zhu et al. “Energy-Efficient Deep Reinforced Traffic Grooming in Elastic Optical Networks for Cloud-Fog Computing”. In: *IEEE Internet of Things Journal* 8 (15 Aug. 2021), pp. 12410–12421. ISSN: 23274662. DOI: 10.1109/JIOT.2021.3063471.

[140] Sha Zhu, Kaoru Ota, and Mianxiong Dong. “Green AI for IIoT: Energy Efficient Intelligent Edge Computing for Industrial Internet of Things”. In: *IEEE Transactions on Green Communications and Networking* 6 (1 Mar. 2022), pp. 79–88. ISSN: 24732400. DOI: 10.1109/TGCN.2021.3100622.

[141] Xunyu Zhu et al. “A Survey on Model Compression for Large Language Models”. In: (Aug. 2023). URL: http://arxiv.org/abs/2308.07633.

A

Results

This index discusses the index of the reproducibility package containing the resulting data.

Reproducability package

Table A.1 shows an index of the measurement, analysis and simulation scripts and the results of this study provided in the reproducibility package as described in subsection 3.2.3

Table A.1: Index of reproducibility package.

Directory/file	Documentation
analyse	
analyse/data	contains all raw and preprocessed data
analyse/results	contains all test results and plots
analyse/analyse_throughput.py	plots Figure 4.17
analyse/calculate_average.py	calculates average for normalisation
analyse/plot_energy_consumption.py	plots energy consumption graphs Figure 4.7-4.18
analyse/preprocess.py	preprocesses measurements into energy per request
analyse/statistical_test.py	performs all statistical tests
llama.cpp	
llama.cpp/dataset	contains dataset used in experiments
llama.cpp/dataset/determine_throughput.py	small sample test determining average throughput
llama.cpp/data/preprocess.py	preprocesses dataset to filter out too large queries
llama.cpp/data/request.py	main script that uses dataset to infer server for experiments
llama.cpp/inference	clone of llama.cpp library for inference server
llama.cpp/measure_jetson/jetson_stats/measure_stats.py	measure script for Jetson
llama.cpp/Containerfile.[device]	respective containerfile for each device
llama.cpp/docker-compose.[device].yml	respective composefile for each device
llama.cpp/docker-compose.throughput[.gpu].yml	composefiles for throughput test depending on hardware availability
llama.cpp/run[_X].sh	run scripts that trigger multiple containers after each other
llama.cpp/test[_X].sh	test scripts
simulate/	
simulate/average_ec.py	retrieves energy consumption averages for simulation
simualte/simulate.py	performs and plots simulations of Figure 4.19-4.20

B

Justification

This appendix discusses some justifications for this study.

Energy justification

To create awareness of the environmental impact of the studies that investigate environmental impacts, we report the energy justifications of this study to inform any other research that wants to reproduce this study or perform similar ones, what the expected energy costs are. To execute this experiment, this study has used a fair amount of energy, even though some steps were taken to reduce this as much as possible. The models used were pre-trained and pre-quantized, which means that those costs were mostly mitigated. We estimate the energy consumption of this project and the related carbon footprint and show it for full disclosure in Table B.1. We divide the usage in the energy consumed for running the experiments, simulating, performing the analysis, using AI for research and development, and weekly video conferencing with supervisors. We acknowledge the fact that these estimations are not very accurate but act more as an indication of the scale of the research.

Table B.1: Disclosure of used energy for this study.

Description	**Calculation**	**Energy Consumption (kJ)**
Experiments	4 envs., 14 quants, 5 thrs. @ avg. 2600J cloud, 500J edge	2300
Simulation	3 + 4 @ ~100J	1
Analysis	4x13x5 @ ~200J	52
AI use	~100x request LLM @ 2000J per request [98]	200
Video conferencing	~30x 30min @ 750J per 3 min [125]	225
Total		**2.778**