On Energy-aware and Verifiable Benchmarking of Big Data Processing targeting AI Pipelines

Georgios Theodorou
UBITECH Ltd
Limassol, Cyprus
0000-0001-5413-2189

Sophia Karagiorgou
UBITECH Ltd
Limassol, Cyprus
0000-0002-1099-8463

Christos Kotronis
UBITECH Ltd
Limassol, Cyprus
0000-0003-2048-8982

***Abstract*—As Artificial Intelligence (AI) is revolutionizing various industries and applications, understanding the hardware requirements and energy consumption of AI pipelines in Big Data (BD) applications has become increasingly essential. This paper presents a comprehensive, scalable framework, designed to systematically measure hardware resources, energy usage, and model performance across two prominent data modalities: tabular data and images. The framework is generalizable, facilitating replicability across the AI research community, and encourages the deployment of AI models with comprehensive metrics beyond traditional accuracy, promoting the optimization of pipelines for real-world scenarios. Through detailed benchmarking, we identify EfficientNet as a standout model for image classification, and XGBoost for tabular data, both excelling in their respective domains. Notably, our findings show that Graphics Processing Units (GPUs) account for approximately 90% of total energy consumption in image-based tasks, while Central Processing Units (CPUs) are responsible for around 50% of energy use in tabular data processing. The merit of our innovative proposed framework combines information theory and probability theory to enhance our understanding of AI model performance in Edge-to-Cloud (E2C) applications that demand efficient Big Data processing in distributed environments. By seamlessly integrating energy efficiency with hardware optimization, it enables real-time monitoring of energy consumption and computing resources in containerized environments, providing precise insights for optimizing AI workloads. This framework facilitates scalable AI deployment on resource-constrained edge devices, reducing energy consumption while enhancing AI model robustness and interpretability, thereby promoting greater trust and transparency in AI-powered decision-making for critical real-world applications. This emphasizes the importance of multi-objective optimization for more sustainable and efficient Big Data AI workflows.**

***Index Terms*—Big Data AI pipelines' benchmarking, AI theoretical framework for edge-to-cloud applications, energy-aware AI pipelines for optimal model placement**

I. Introduction

Edge-to-Cloud (E2C) applications, particularly in the context of Artificial Intelligence (AI) system development, deployment, and serving are often treated as Big Data (BD) problems with regard to training tasks due to the complexity of Neural Network (NN) architectures. AI engineers, practitioners, researchers, and solution providers are typically concerned with the computational resources consumed by AI training tasks and the frequency of performed AI inference tasks. The situation becomes more complex when energy-constrained devices such as drones, Virtual Reality (VR) and Augmented Reality (AR) glasses, and mobile devices are used. For both AI training and inference tasks, it is essential for users to consider the computational running time, the size of the AI model along with its parameters, energy efficiency, and the quality of the results.

Critical application metrics such as energy consumption, the complexity of NN architectures, and the optimization of Central Processing Unit (CPU), Graphics Processing Unit (GPU), and Random-Access Memory (RAM) are often overlooked during the development, training, and inference of AI models. This oversight can largely be attributed to the lack of comprehensive benchmarking frameworks that consider the trade-offs between contradictory axes, such as energy consumption versus AI model accuracy, NN model architecture complexity versus epochs or training time, etc. Most comparative studies in AI development focus on the models' performance metrics, including accuracy, precision, loss, and error, with little attention paid to underlying hardware considerations, including CPU, GPU, and RAM utilization, energy consumption, and inference frequency. This is a shortcoming, as the training and inference of AI pipelines in E2C applications must also consider field conditions, energy constraints of the devices regarding their optimal placement, and the criticality of decision-making that dictates higher model precision.

Nowadays, AI systems are becoming increasingly pervasive across different industries and applications, and the significance of these metrics cannot be overstated. Understanding the energy footprint of AI models is essential not only from an environmental sustainability perspective but also from an economic standpoint. As AI deployments scale, the ability to anticipate energy consumption and computing costs during model development can lead to more efficient and responsible resource utilization, optimal model placement (whether at the edge or in the cloud), and improved model optimisation. In addition, application criticality is a key factor when energy consumption is a primary concern (e.g., mobile devices, drones, etc.), or when high reliability and accuracy are required (e.g., medical sensors and diagnostics, etc.), or in cost-sensitive applications where the focus is on finding the optimal balance between performance and energy consumption. Consequently, making informed decisions regarding the choice and optimal placement of AI models can yield significant benefits, including reduced operational costs and lower environmental impact.

Benchmarking energy-aware Big Data AI pipelines ensures the correctness and reproducibility of these pipelines, the exploration of trade-offs between different pipeline architectures and hardware configurations, and the optimization of energy usage without compromising performance for critical applications.

The latest State-of-the-Art (SotA) research [1] highlights the need to enhance the capabilities of Big Data analytics benchmarking. This will improve the reliability, performance, and operational efficiency of applications. The target audience for benchmarking Big Data AI pipelines is broad, including data scientists and machine learning engineers / practitioners, responsible for developing and deploying AI pipelines, who need to evaluate their performance in order to optimize them, as well as mission-critical application providers who rely on benchmarking to compare different placements, approaches, and algorithms. The merit of benchmarking Big Data AI pipelines in E2C application contexts is twofold. Firstly, it allows for the verification of the accuracy and reliability of a pipeline's results, ensuring that the AI model produces correct and consistent outputs. Secondly, it considers resource allocation, energy consumption, data efficiency, and optimal placement.

This work introduces a scalable and fully generalisable theoretical and software solution that enables the measurement, persistence, and visualization of energy consumption, hardware resource utilization, and AI model performance efficiency during the training phase. The proposed approach leverages containerization to encapsulate AI model training phases and deployment within a Kubernetes cluster. The cluster is complemented by a suite of tools designed to monitor and visualize the defined metrics, including energy usage, AI models' performance efficiency, and CPU / GPU / RAM allocation. In order to abstract and generalise the derivatives, we have conducted experiments with several widely-used AI models across two different data modalities, namely tabular and imagery data. Furthermore, the mention of GPUs is only relevant in the context of image modalities, as not all tabular data APIs, such as scikit-learn, support GPU acceleration.

The contributions of this paper are as follows:

- Tackling Big Data AI pipeline efficiency and energy preservation, providing abundant explanations on AI model selection, rationale, and optimal placement.
- Applying a meta-algorithmic approach to improve AI model performance, and introducing a novel solution that extends the findings with explanations with confidence indicators, illustrating key attributes that contribute to benchmarking conclusions.
- Encouraging a shift within the AI community towards deploying AI models with rich justifications about various metrics to meet multiple or even contradictory objectives.
- Facilitating end-users to optimally select and optimise Big Data AI pipelines for real-world applications, while considering resource constraints.

The rest of the paper is organized as follows: Section II provides current literature review on benchmarking, performance tuning, and optimisation for Big Data applications. Section III introduces the theoretical framework for decomposing Big Data AI pipelines and measuring their behavior in edge and cloud computing infrastructures. Section IV details the technical architecture and software solution for benchmarking the Big Data AI pipelines in E2C applications. Section V discusses our experimental results, while Section VI concludes the paper and outlines future directions.

II. Literature Review

A comprehensive study on the aspects that affect the Big Data AI pipeline training, considering different objectives, highlights significant variations in both performance and energy consumption during Deep Neural Networks (DNN) training [2]. Both the system architecture, including CPUs, GPUs, and Tensor Processing Units (TPUs), and the AI model complexity should be considered during benchmarking when optimising the training phase. While GPUs and TPUs provide high throughput for tasks such as image recognition and speech-to-text, energy efficiency can greatly differ depending on the hardware and the optimization techniques applied. These findings underscore the necessity of considering both performance and energy consumption when selecting hardware for AI training, particularly in environments where cost, energy preservation, and efficiency are critical factors.

The importance of evaluating energy consumption in machine learning (ML) is widely recognized for monitoring, understanding, and optimizing its computational and environmental impact. However, there is no single approach that can address all use cases, and there is an ongoing debate about the best methods to evaluate energy consumption for specific applications. In the meantime, various methods, each with unique strengths and limitations, have been developed. A systematic review of these approaches, designed to evaluate energy consumption during both training and inference, was conducted, followed by an experimental protocol to compare the effectiveness of these methods across diverse AI tasks, including vision and language models [3].

Several libraries have emerged to address the challenge of tracking energy consumption in AI pipelines. One prominent example is the eco2AI library [4], which offers a powerful solution for monitoring energy usage and CO2 emissions during both training and inference phases. This library tracks CPU, GPU, and RAM utilization by gathering power consumption logs through process identification (PID) and system metrics retrieved using Linux commands, such as *top*.

Similarly, the EfiMon tool [5] provides a granular, non-invasive method for tracking energy consumption at the process level. EfiMon uses regression-based models to estimate energy usage with high precision, even in shared computing environments. This tool has shown small deviations in its measurements on Intel and AMD systems, making it a valuable resource for optimizing energy consumption in AI research and high-performance computing (HPC) [6].

EIT [7] is another tool that simplifies real-time monitoring of energy usage and carbon emissions during AI training. This tool facilitates the generation of standardized online reports

and leaderboards, promoting responsible research practices, especially in the context of energy-efficient reinforcement learning algorithms.

CarbonTracker (CT) [8] is a specialized tool for tracking the carbon emissions produced during AI model training. This framework helps researchers measure the environmental impact of their models, offering valuable insights for developing more sustainable AI systems.

At the same time, there has been an evident revolution in the industry thanks to the improvement of AI pipelines with a high level of accuracy performance even surpassing human capabilities for different sorts of problems. Achieving a high accuracy rate is closely related to models that have a vast amount (millions or even billions) of weights (parameters) that are supposed to contain the information learnt from training data. However, many modern AI methods have a black-box nature, which hinders their adoption by practitioners in many application fields. This issue raises a recent emergence of a new research area in AI, setting the ground for (i) extensive benchmarking over different algorithms and methods to understand their behaviour across different data modalities; (ii) use of data and features of different granularity and veracity to optimise the learning capability and thus the performance of an AI model; and (iii) monitoring the underlying compute resources to dimension the financial, computing or energy costs of AI model training and to derive trade-offs (i.e. through what-if analyses) regarding smart placement, energy consumption, and other business-defined objectives. In addition, there is the need to explain the behaviour of an AI model, aiming at providing more understandable, interpretable, and justifiable for humans AI-based decision-making processes and outcomes. Several theoretical frameworks for AI have been introduced to tackle different directions, focusing on (i) Explainable Artificial Intelligence (XAI); (ii) Emergent Behaviour and Alignment of AI; (iii) AI as Originator and Facilitator of Innovation; and (iv) Three-Level Model for AI while Learning.

Tchuente et al. [9] proposed a new methodological and theoretical framework for XAI decomposed into six steps that can be followed by all practitioners or stakeholders to improve the implementation and adoption of XAI in their business applications. They highlighted the need to rely on domain field and analytical theories to explain the entire analytical process, from the relevance of the business question to the robustness checking and validation of explanations provided by XAI methods.

Rizzo et al. [10] fit explanations of an AI model into the properties of faithfulness (i.e. the explanation is an accurate description of the model's inner workings and decision-making process) and plausibility (i.e. how much the explanation seems convincing to the user). Their theoretical framework simplifies the operationalization of these properties, and provides new insights into common explanation methods that they analyse as case studies. They also discuss the impact of their framework in biomedicine, a very sensitive application domain where XAI can have a central role in generating trust.

Freund et al. [11] explore the complex dynamics of emergent behaviour and alignment within AI systems and present a comprehensive framework for conceptualizing and modeling these phenomena. Their framework incorporates the multi-level and time-dependent nature of emergent behaviour and alignment, considering the interplay between system states, inputs, function rules, learning algorithms, environments, and historical data. The proposed framework sheds light on the challenges and opportunities associated with achieving and maintaining alignment in AI systems.

Brem et al. [12] introduced a two-part conceptual AI framework: The first part views AI as a technology that can fulfill different roles within a company, and the second looks at AI and its use along the company's innovation processes. They also discussed these two views using examples from existing field applications and described potential areas for future research and limitations of the proposed framework.

Gibson et al. [13] introduced a three-level model that synthesizes and unifies existing learning theories to model the roles of AI in promoting learning processes. The model, drawn from developmental psychology, computational biology, instructional design, cognitive science, complexity, and sociocultural theory, includes a causal learning mechanism that explains how learning occurs and works across micro, meso, and macro levels. The model also explains how information gained through learning is aggregated, or brought together, as well as dissipated, or released and used within and across the levels.

Last, Haidar [14] proposes a novel integrative theoretical framework for Responsible AI (RAI), which addresses four key dimensions: technical, sustainable development, responsible innovation management, and legislation. The responsible innovation management and the legal dimensions form the foundational layers of the framework. The first embeds elements like anticipation and reflexivity into corporate culture, and the latter examines AI-specific laws from the European Union and the United States, providing a comparative perspective on legal frameworks governing AI. The study's findings are helpful for businesses seeking to responsibly integrate AI, developers who focus on creating responsibly compliant AI, and policymakers looking to foster awareness and develop guidelines for RAI.

Within this expansive domain of Big Data and edge computing, AI stands as a beacon, transforming raw data into actionable insights and automating a myriad of complex tasks. However, the intricate relationship between AI and Big Data gives rise to various technical challenges, like the number of training epochs and time, over-/under-fitting, and data leakage, which can influence the efficacy of AI models. Last, the inherent characteristics of AI models and the energy-constrained edge devices further contribute to technical challenges while optimizing AI models for smart placement, cost, energy reduction, and more.

This work introduces a novel AI theoretical framework for understanding and evaluating the behavior and operation of AI methods in E2C execution contexts. We take into account two

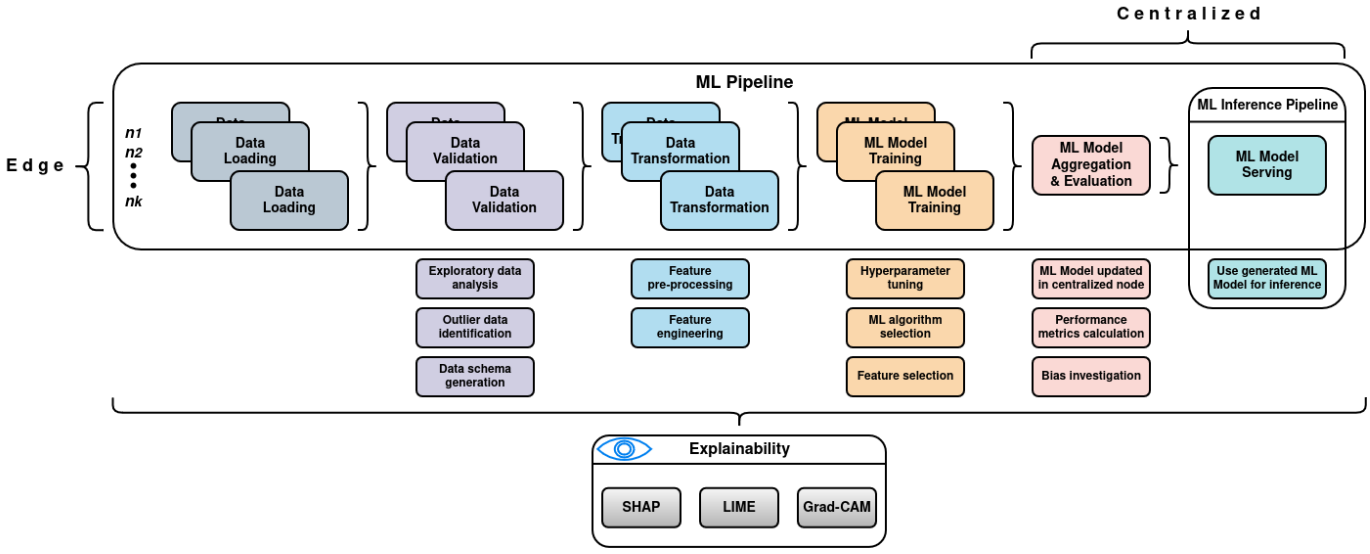

Fig. 1. AI Theoretical Framework

key axes: information theory and probability theory. Through information theory, we investigate how the complexity and information gain of AI models affect their compute, energy, cost, and performance metrics in the E2C contexts. Through probability theory and statistical analysis, we analyse the likelihood of different AI models' results and quantify their confidence levels and potential for errors. To the best of our knowledge, this work is the first to introduce both a theoretical and software framework that monitors AI applications within a containerized environment, capturing real-time energy consumption and computing resources (e.g., CPU, RAM, number of threads). The containerization paradigm allows for a more modular, scalable, and fine-grained approach to tracking energy usage compared to traditional process-level (PID) monitoring. This approach allows for better isolation, reproducibility, and consistency, as containerized applications abstract individual system processes and interact with the underlying hardware in a more holistic and integrated manner. As a result, the proposed approach ensures more accurate insights into the performance efficiency, energy, and computing consumption of AI workloads, especially in complex, resource-constrained edge devices and compute-shared environments, like Kubernetes clusters.

III. AI Theoretical Framework

This section presents a novel approach, AI Theoretical Framework Fig. 1, to structuring an AI training pipeline by splitting it into discrete steps, applicable regardless of data modality or execution mode—whether federated, centralized, edge, or cloud. This framework offers several advantages, including a clear definition of the various functions that need to be executed at each step and within each compute environment context. Given the inherent complexity of AI pipelines, this approach provides much-needed clarity and guidance, creating a seamless conceptualization and development process.

The first step in a Big Data AI pipeline is Data Loading, where the data must be efficiently and optimally loaded to ensure that sufficient memory is available and that parsing during subsequent steps is smooth. State-of-the-art techniques for data loading include utilizing multi-threaded data pipelines, caching mechanisms, and pre-fetching, to reduce data access latency. For instance, frameworks like TensorFlow's tf.data API and PyTorch's DataLoader implement parallelized loading, allowing for effective utilization of available hardware resources. This is particularly important in distributed training environments, where efficient data loading can help avoid bottlenecks and ensure that GPUs and CPUs operate at their full potential.

Key considerations at this step involve balancing data input/output (I/O) operations, ensuring hardware resources are not underutilized, and performing memory-efficient data transformations such as batching and sharding to avoid memory overflow in large datasets. These techniques enable the system to scale efficiently in edge or cloud environments, making it adaptable to various data modalities.

The second step is Data Validation, where the quality of the data is critically assessed, and any errors, outliers, or irrelevant data are identified and corrected. This phase also includes Exploratory Data Analysis (EDA), which provides initial insights into the dataset. For tabular data, common validation techniques involve checking for missing values, normalizing distributions, and identifying outliers using statistical methods like z-scores or interquartile range (IQR). For imagery data, techniques such as detecting corrupted images, checking image dimensions, or ensuring consistency in file formats are critical.

State-of-the-art methodologies include automated data validation libraries such as Great Expectations and Pandas Profiling, which help automate some of these checks. For image modalities, libraries like Albumentations and tslearn provide effective preprocessing techniques to ensure high-quality data

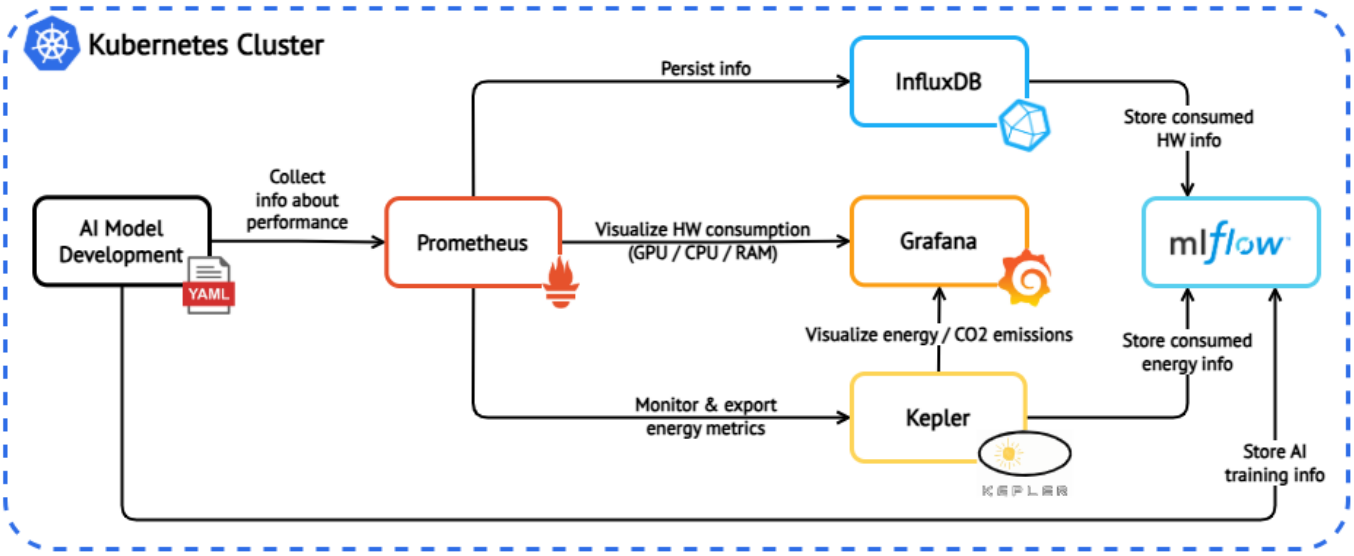

Fig. 2. Visualization of the Benchmarking Framework

is available for AI model training.

In the Data Transformation step, features are engineered to enhance the predictive power of models. In classical machine learning, feature engineering may involve encoding categorical variables, normalizing numerical data, or creating interaction terms. For example, techniques like one-hot encoding, standardization, and dimensionality reduction (PCA, t-SNE) are commonly used in tabular datasets.

In neural networks, especially for deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), data transformation often involves preparing inputs by resizing images, normalizing pixel values, or transforming sequential data to fit the architecture's input format. Data augmentation techniques, such as flipping, cropping, or rotation for images, also play a critical role in increasing the diversity of the training data without the need for additional data collection. These transformations help ensure that models generalize better and are robust afainst overfitting.

The fourth step is Model Training, where the preprocessed data is fed into the chosen model architecture. Depending on the learning task, this could be a classical machine learning model such as Random Forest or Support Vector Machines (SVMs) or a deep learning model such as CNNs or Transformers. Critical decisions at this stage include selecting optimization algorithms (e.g., Adam, Stochastic Gradient Descent - SGD), choosing loss functions (e.g., cross-entropy for classification, mean squared error for regression), and determining the best hyperparameters through techniques like Grid Search or Bayesian Optimization. For neural networks, additional considerations such as determining the number of epochs, batch size, learning rate scheduling, and dropout are essential for achieving efficient training. In distributed or federated learning setups, models are trained across multiple clients, nodes, or devices, and this distributed architecture should be carefully managed to optimize resource usage and minimize latency.

Finally, the Model Evaluation step involves testing the model on a validation or test set to assess its performance based on metrics such as accuracy, precision, recall, or mean absolute error, depending on the task. In centralized approaches, evaluation is straightforward; however, in Federated Learning, the evaluation takes on additional complexity. Models trained on different clients / nodes must be aggregated on a central server. Aggregation techniques such as Federated Averaging (FedAvg) ensure that model parameters from different nodes are combined to form a global model. Post-aggregation testing ensures that the global model satisfies the required accuracy or performance benchmarks.

By decomposing and abstracting a Big Data AI pipeline into discrete steps, the AI Theoretical Framework not only allows for a clear roadmap for implementation but also enables flexibility across execution modes, whether federated or centralized, on edge devices, or in the cloud. This modular approach supports scalable, responsible, and resource-aware AI development, ensuring that performance tuning and optimization can be systematically applied throughout the pipeline.

IV. Experimental Framework

The experimental framework, as illustrated in Figure 2, effectively captures the metrics of energy consumption, performance efficiency, and hardware utilization. This framework is hardware and platform vendor-agnostic, designed to cover a broad range of scenarios and use cases. We place particular emphasis on ensuring our framework is both highly scalable and extensive, making it practical for real-world Edge-to-Cloud deployments. By tracking system operations on a per-second basis, our framework effectively captures dynamic workloads and fluctuating resource availability, even

in cases of abrupt changes. However, we record the technical specifications of the target hardware and platform based on their objective capacity. The first step includes the abstraction and containerization of the Big Data AI pipeline within a Kubernetes cluster [15]. Proper configuration and setup of the cluster are necessary for monitoring the execution environment of the Big Data application under evaluation.

The Kubernetes cluster supports E2C applications and provides scalability, easy integration of edge devices, and dynamic resource allocation. We study the case of Big Data AI pipelines that require dynamic resource adaptation, as the training and inference phases may experience random fluctuations. In addition, containerization within Kubernetes ensures process isolation and minimizes interference from other processes (e.g., from the operating system or neighboring applications), leading to more accurate and reproducible measurements of energy and computing resource consumption. Moreover, Kubernetes' native tools for monitoring, such as Prometheus and Grafana, allow for real-time tracking and visualization of hardware metrics (CPU, GPU, RAM) and energy usage of a given AI pipeline. This distributed approach enables the monitoring of federated workloads across multiple nodes, offering a more comprehensive and granular insight into the system's performance compared to traditional PID-based monitoring. Lastly, leveraging Kubernetes' cloud-native infrastructure, the framework can be easily extended to larger, multiple, and more complex environments, supporting diverse AI applications and varying computational demands.

We have deployed and integrated a comprehensive suite of libraries and tools to form the monitoring infrastructure, ensuring precise tracking of energy consumption and hardware utilization during the various steps of a Big Data AI pipeline. Each of these components has been carefully selected for its ability to address specific challenges in resource monitoring. In the following, we provide a brief overview of the tools employed, offering necessary background for those unfamiliar with them, and explain their role within the overall architecture.

Prometheus [16], integrated into the Kubernetes cluster, serves as the primary monitoring tool to collect real-time hardware utilization data. It tracks vital system metrics, including CPU, GPU, and RAM usage, and provides a highly reliable and scalable method for capturing these metrics. By integrating with the Kubernetes environment, Prometheus ensures continuous monitoring, accurately recording resource fluctuations over time. Moreover, Prometheus gathers a wide variety of statistics on top of the application it monitors, providing high flexibility for any use case-specific monitoring requirements, such as information on disk throughput, filesystem I/O, and out-of-memory (OOM) errors.

Once the hardware utilization data is captured by Prometheus, it is stored in InfluxDB [17], a high-performance time-series database chosen for its ability to persist and handle large volumes of time-stamped data for long periods. InfluxDB ensures long-term data storage and facilitates efficient querying and analysis of historical hardware consumption trends. This long-term storage is crucial for monitoring the energy consumption patterns of AI applications over extended periods, particularly in Big Data scenarios where these patterns evolve.

For visualization, Grafana [18] is employed to create intuitive and interactive dashboards. We make use of two purposefully developed Grafana dashboard templates, [19] for visualization of HW consumption and [20] for visualization of energy and CO2 emissions, that enables Grafana to connect and pull data from Prometheus. This visualization capability allows developers and researchers to gain real-time insights into the AI application's resource usage, enabling them to monitor trends and make informed decisions regarding optimizations, as well as optimal resource allocation and scheduling.

Kepler [21] extends the capabilities of Prometheus by leveraging system telemetry data to compute energy consumption and carbon emissions. Kepler integrates seamlessly with Prometheus, offering real-time insights into the environmental impact of the AI application, including power usage and CO2 emissions. This promotes sustainable computing practices, offering both technical and environmental metrics that are critical for optimizing energy consumption in containerized AI workloads.

Finally, the AI models, their performance efficiency, and any associated metadata are stored and managed using MLFlow [22]. MLFlow enables reproducibility by tracking experiment runs, saving model versions, and maintaining all relevant information for future reference. This ensures that model performance, energy consumption, and hardware usage can be monitored and compared across different experiments, providing a complete lifecycle management system for AI development through the proposed framework.

V. Benchmarking Results

We conducted an extensive comparative analysis using both the theoretical and software frameworks described above, focusing on two distinct data modalities, i.e., images and tabular data. These modalities represent a broad range of machine learning tasks, commonly encountered in real-world applications, making them ideal for this study. By selecting the most widely used algorithms for each modality, we performed benchmarks that are highly relevant to a broad audience, from researchers to solution providers.

For each modality, we utilized publicly available datasets to ensure consistency in model evaluation and comparison. By focusing on well-known datasets, the goal was to control for dataset-specific variability and instead highlight the comparative performance of the algorithms. This approach allowed us to better assess AI model efficiency, hardware utilization, and energy consumption across different learning tasks, rather than being influenced by the characteristics of specific datasets. Researchers and practitioners can seamlessly extend the framework to incorporate additional datasets, including custom or domain-specific data while maintaining the robustness of the comparative analysis.

Furthermore, the abstracted design of the framework ensures its adaptability to other machine learning tasks beyond the

two data modalities we tested. This is realized by abstracting every piece of computation through Kubernetes pods, thus allowing for easy integration of new AI models. This enables researchers to benchmark their models against established baselines efficiently. The scalability of this approach is especially important as machine learning applications continue to diversify, and the need for flexible, generalizable benchmarking frameworks becomes more critical. As a result, the framework not only serves the immediate purpose of this study but also provides a valuable tool for ongoing research and development in machine learning and AI energy-aware analysis.

All experiments were conducted on a consistent system, which is described in detail by the hardware specifications in Table I. This setup ensures uniformity across all trials and allows for controlled testing conditions, which is essential for valid comparisons and results.

HW Component	Capacity
CPU	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
GPU	NVIDIA GeForce RTX 2070
RAM	32GB

TABLE I
HARDWARE SPECIFICATIONS OF THE TARGET SYSTEM

In the following tables, where the results of the experiments are presented, it is important to note that the average number of threads (Avg. Threads) is measured by Prometheus and refers to the number of threads associated with a container as collected by cAdvisor or Kubernetes' container metrics exporter. It offers valuable insights into the implementation of AI algorithms.

A. Tabular Modality

For the tabular data experiments, we conducted an in-depth evaluation of four widely-used classical machine learning models: Categorical Boosting [23], Gradient Boosting [24], LightGBM [25], and Extreme Gradient Boosting [26]. Random Forest [27] could not be successfully benchmarked due to an "Out of Memory" error encountered with our dataset, indicating significantly higher RAM usage compared to other models. This highlights the potential limitations of Random Forest for Big Data applications, particularly in resource-constrained environments. The dataset was sourced from a Kaggle competition featuring the top 50 Spotify songs in 2019 [28], providing a robust foundation for comparative analysis. The dataset consisted initially of 50K records and 20 features and was artificially increased to 4.4 million records, amounting to a total size of 767 MB. Each model was trained using hyperparameter tuning via a random Grid Search approach, ensuring optimal model configurations and enabling a fair and rigorous comparison of the algorithms.

It is important to emphasize that the focus of this study was not on identifying the most effective model for predicting which Spotify songs would enter the top 50, but rather on benchmarking various performance objectives of these algorithms using a well-known and standard dataset. By maintaining this focus, we further compared the models in terms of performance efficiency, energy consumption, and hardware utilization under common conditions. This approach allows us to provide insights into how each model performs in a typical tabular data scenario, and how model complexity or envisioned prediction accuracy affects energy, optimal placement, and hardware utilization.

For the implementation of Random Forest and Gradient Boosting, we utilized the well-established scikit-learn library [29]. The LightGBM model was implemented following the official framework described in [30]. For Categorical Boosting (CatBoost), we adopted the implementation provided by the CatBoost library [31], and the Extreme Gradient Boosting (XGBoost) model was implemented using the official XGBoost documentation [32].

As shown in Table II, XGBoost delivers an optimal balance between execution time and precision, completing the task significantly faster than the other models while maintaining impressive accuracy. Its execution time makes it highly suitable for scenarios where speed and efficiency are crucial. Surprisingly, LightGBM, which is often praised for its speed, was significantly slower than expected, taking more than twice as long as the other models to complete the task.

Table III provides further insight into the energy consumption patterns of the AI models under evaluation. XGBoost not only finishes the task in the shortest amount of time but also demonstrates remarkable energy efficiency, consuming significantly less energy compared to the other models. In stark contrast, LightGBM (LGBM) emerges as the least efficient model, consuming 27 times more energy than XGBoost. This substantial disparity in energy consumption highlights the complexity differentiation in the underlying architectures and their efficiency in handling computational tasks.

An interesting observation is that XGBoost, being both the fastest and the most energy-efficient model, also exhibits the lowest CPU and RAM utilization. This suggests that XGBoost is optimized for resource management, maintaining high performance while minimizing its hardware footprint. Conversely, LGBM, the least energy-efficient model, places the heaviest demand on CPU resources, especially through its extensive use of multithreading.

The intensive CPU utilization in LGBM accounts for excessive energy consumption and longer execution time. Therefore, an intriguing trend can be observed: energy consumption appears to closely correlate with CPU usage. AI models that utilize less CPU power consume less energy and, correspondingly, are less memory-intensive. This relationship underscores the importance of CPU efficiency in determining overall energy consumption, making it a crucial factor when optimizing AI models for large-scale tasks in energy-constrained environments.

B. Images Modality

For the image modality experiments, we selected 8 of the most commonly used deep learning models in the field of computer vision: ConvNext [33], DenseNet [34], EfficientNet

AI Model	Modality	Records	Size (MB)	Accuracy	Training Time (mins)
CatBoost	Tabular	4.4M	767MB	76.9%	45
LGBM	Tabular	4.4M	767MB	73.5%	155
Gradient Boosting	Tabular	4.4M	767MB	77.5%	70
XGBoost	Tabular	4.4M	767MB	77.1%	7.35

TABLE II
COMPARISON OF TABULAR DATA FOR AI MODELS PERFORMANCE AND TRAINING TIME

AI Model	Energy Consumed (Joules)	Avg CPU Usage %	Avg Memory Usage (GB)	Avg Threads	CPU Energy Impact
CatBoost	124,544	406%	17	150	50%
LGBM	488,931	484%	14	97	44.2%
Gradient Boosting	162,169	450%	13	56	54.2%
XGBoost	17,451	368%	11	42	31%

TABLE III
COMPARISON OF TABULAR DATA FOR AI MODELS HARDWARE AND ENERGY CONSUMPTION

[35], MobileNet [36], ResNet [37], VGGNet [38], Vision Transformer [39] and YOLO [40]. These models were chosen for their wide applicability in both academic research and real-world engineering solutions. All models, except YOLO, were implemented using the torchvision library, while YOLOv8 was sourced from the Ultralytics open-source work. Using these established implementations ensures that we leverage optimized and widely accepted versions of each model, making our results more generalizable and reliable. The task under study is an image classification learning task, with the CIFAR-10 dataset [41] serving as the underlying dataset. CIFAR-10 is a well-known and frequently used dataset for image classification tasks, consisting of 60,000 images across 10 different classes, with a total dataset size of 163 MB. This makes it a suitable dataset for comparing model performance across various metrics.

As with the tabular data modality, the objective of this study was not to achieve the highest accuracy on the CIFAR-10 dataset—numerous approaches have already demonstrated near-perfect classification performance. Instead, we focused on conducting a fair and rigorous comparison of these models for hardware utilization, energy consumption, and execution time. To ensure consistency across evaluations, each AI model was trained for a fixed number of 4 epochs. Moreover, the models were not pre-trained; instead, each model was trained from scratch, retaining only the architecture, with their initial weights set to None. This approach ensured that all models started from the same baseline, eliminating any potential advantages that could arise from transfer learning or pre-training on similar datasets. This allows us to directly compare the computational efficiency and resource consumption of each model, providing valuable insights into how different architectures handle the same task under identical conditions.

As shown in Table IV, EfficientNet is the most efficient network, achieving the highest precision, accuracy, and F1 score across the four training epochs. It outperforms the second-best model, VGGNet, by nearly 10% in accuracy, while completing the task in a significantly shorter time frame and using 73% less energy. This remarkable efficiency in both time and energy consumption highlights EfficientNet's better architecture in balancing performance and resource utilization.

Interestingly, while MobileNet completes the training the fastest, EfficientNet offers a better combination of accuracy and energy efficiency. The comparison reveals an interesting trend where AI models with either a relatively small number of parameters, like EfficientNet and MobileNet, or a large number of parameters, such as VGGNet, Vision Transformer, and ResNet152, outperform those with a medium number of weights, such as ConvNext and YOLOv8. This observation suggests that small networks are well-suited for scenarios where quick execution and energy-aware placement are crucial. In contrast, medium-sized networks may struggle to balance speed and performance effectively, justifying the preference for smaller architectures in time-sensitive or energy-constrained application cases.

As depicted in Table V, MobileNet demonstrated the lowest energy consumption, requiring approximately 32K Joules, followed closely by EfficientNet. In contrast, the most energy-intensive networks were Vision Transformer and YOLOv8, with YOLOv8 being particularly noteworthy for its subpar performance relative to its energy usage, making it the least optimal model among those evaluated.

Interestingly, the number of parameters does not appear to be the primary driver of energy consumption; instead, the time required for execution plays a more significant role. For instance, despite ConvNext having more than three times the number of parameters as DenseNet, the two models completed their training at nearly the same time and exhibited very similar energy consumption levels. This suggests that model architecture and execution efficiency have a more pronounced impact on energy usage than the sheer number of parameters.

Moreover, an intriguing observation is that YOLOv8, unlike the other models, exhibited significantly higher RAM and CPU utilization. This points to a potential inefficiency in resource allocation, particularly when considering its lower performance in comparison to the other networks. These findings underscore the importance of not only evaluating accuracy but also considering resource efficiency when selecting models for deployment in energy-constrained environments.

It was observed that, on average, 90% of the total energy consumed by the networks could be attributed to GPU utilization, with the smaller networks having a ratio of 82%

AI Model	Modality	Records	Size (MB)	Num. Weights	Precision	Accuracy	F1 Score	Training Time (sec)
ConvNext (Tiny)	Images	60K	163MB	28.6M	32.5%	30%	26%	1197
DenseNet121	Images	60K	163MB	8M	64%	63%	63%	1128
EfficientNetb0	Images	60K	163MB	5.3M	72%	71.5%	71.5%	596
MobileNet(v2)	Images	60K	163MB	3.5M	63%	63%	62.6%	371
ResNet152	Images	60K	163MB	60.2M	54%	52%	51%	2467
VGGNet16	Images	60K	163MB	138M	66%	65%	64%	2199
Vision Transformer (Base)	Images	60K	163MB	86M	59%	56%	55.8%	2577
YOLOv8	Images	60K	163MB	27.3M	38%	37%	33%	2641

TABLE IV
COMPARISON OF IMAGERY DATA FOR AI MODELS PERFORMANCE AND TRAINING TIME

AI Model	Energy Consumed (Joules)	Avg CPU Usage %	Avg Memory Usage (GB)	Avg Threads
ConvNext (Tiny)	79248	80.25%	0.76	12
DenseNet121	74445	91.78%	1.45	16
EfficientNetb0	39331	106.3%	0.78	11
MobileNet(v2)	32268	149%	1.17	13
ResNet152	159031	79%	0.91	14
VGGNet16	135813	101.28%	0.89	15
Vision Transformer (Base)	169813	103%	1.02	16
YOLOv8	167321	144%	6.89	90

TABLE V
COMPARISON OF IMAGERY DATA FOR AI MODELS HARDWARE AND ENERGY CONSUMPTION

and bigger ones 90%, underscoring the significant energy demands of GPUs compared to other hardware components such as CPU and RAM. This finding highlights the energy-intensive nature of GPU operations in AI training and suggests a pressing need for further research into optimizing GPU energy efficiency. Addressing this imbalance is critical for reducing the overall energy footprint of AI models, especially as their deployment becomes more widespread across diverse environments, from cloud data centers to edge devices.

The metrics provided exhibit a deviation of a maximum 20%, as observed across multiple experiment runs.

We would like to clarify that benchmarking was also performed for the steps of Data Loading, Validation, Transformation, and Model Evaluation. However, we do not report the results in this paper, as these phases are allocating computing time, and thus consuming energy, which is identical to the data size. The primary focus of this work is to benchmark the Big Data nature of AI pipelines, and most importantly the phase that mostly contributes to greater execution times, energy consumption, and hyperparameters tuning.

The key takeaways of benchmarking on Big Data AI pipelines are summarised as follows:

- Comprehensive, multi-metric benchmarking of leading models across Tabular and Image data modalities.
- Scalable and generalizable framework for benchmarking Big Data applications, incorporating both energy consumption and hardware utilization.
- A theoretical AI framework that provides a detailed, step-by-step analysis of AI pipeline components.

VI. CONCLUSION

The proliferation of AI applications in E2C computing environments has led to a growing demand for efficient and scalable execution. However, understanding the behavior and performance of AI algorithms in these dynamic contexts remains a significant research challenge, requiring empirical modeling, tuning, and optimizing their performance. The proposed theoretical and software frameworks efficiently address both performance and energy preservation of Big Data AI pipelines, providing a detailed analysis of optimal model selection and placement within E2C environments.

Looking ahead, we plan to extend our framework to benchmark leading models in the time-series modality, addressing a gap in the current understanding of time-series performance and resource consumption. Additionally, we aim to develop an open-source tool, enabling easy adoption of our framework by the broader AI community. As new models emerge, we will continue benchmarking not only their accuracy but also their hardware and energy consumption. Ultimately, our goal is to encourage the AI community to shift its focus beyond accuracy alone, considering comprehensive metrics such as energy efficiency and hardware utilization. This approach promotes a more sustainable and economically viable mindset, benefiting both the environment and AI system optimization. We also aim to conduct extensive comparative experiments between our framework and current state-of-the-art tools for energy monitoring in AI applications. This will allow us to assess the accuracy of our framework, highlight potential limitations in existing methodologies, and demonstrate how our proposed solution addresses these issues by improving the accuracy and fidelity of energy monitoring in AI systems. We will place particular emphasis on comparing our framework with existing tools such as the eco2AI library [4], the EfiMon tool [5], EIT [7], and CarbonTracker (CT) [8]. We also aim to conduct a thorough quantitative analysis of the energy consumption trade-offs associated with models optimized through transfer learning and pruning techniques. By evaluating these trade-offs, we seek to understand how these optimization methods influence energy efficiency while maintaining model performance. Specifically, our analysis will assess the extent that transfer learning, with its ability to leverage pre-trained

models, reduces the overall resource demands. Similarly, we will investigate the effectiveness of pruning in eliminating redundant parameters, potentially decreasing overall energy usage.

VII. Acknowledgements

This work has received funding from the European Union's Horizon Europe TALON project under GA No 101070181 and the European Defence Fund 2022 Programme (EDF 2022-DA) PROTEAS project under GA No. 101121371.

References

[1] N. A. Ochuba, D. O. Olutimehin, O. G. Odunaiya, O. T. Soyombo *et al.*, "Reviewing the application of big data analytics in satellite network management to optimize performance and enhance reliability, with implications for future technology developments," *Magna Scientia Advanced Research and Reviews*, vol. 10, no. 2, pp. 111–119, 2024.

[2] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, "Benchmarking the performance and energy efficiency of ai accelerators for ai training," in *Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 744–751.

[3] C. Rodriguez, L. Degioanni, L. Kameni, R. Vidal, and G. Neglia, "Evaluating the energy consumption of machine learning: Systematic literature review and experiments," *arXiv preprint arXiv:2408.15128*, 2024.

[4] S. A. Budennyy, V. D. Lazarev, N. N. Zakharenko, A. N. Korovin, O. A. Plosskaya, D. V. Dimitrov, V. S. Akhripkin, I. V. Pavlov, I. V. Oseledets, and I. S. Barsola, "Eco2ai: Carbon emissions tracking of machine learning models as the first step towards sustainable ai," *Doklady Mathematics*, vol. 106, no. Suppl 1, pp. S118–S128, 2022.

[5] L. G. Leon Vega, N. Tosato, and S. Cozzini, "Efimon: A process analyser for granular power consumption prediction," *arXiv preprint arXiv:2409.17368*, 2024.

[6] L. G. León-Vega, N. Tosato, and S. Cozzini, "Efimon: A process analyser for granular power consumption prediction," *arXiv preprint arXiv:2409.17368*, 2024.

[7] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *arXiv preprint arXiv:2002.05651*, 2020.

[8] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," *arXiv preprint arXiv:2007.03051*, 2020, iCML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems.

[9] D. Tchuente, J. Lonlac, and B. Kamsu-Foguem, "A methodological and theoretical framework for implementing explainable artificial intelligence (xai) in business applications," *Computers in Industry*, vol. 155, p. 104044, 2024.

[10] M. Rizzo, A. Veneri, A. Albarelli, C. Lucchese, M. Nobile, and C. Conati, "A theoretical framework for ai models explainability with application in biomedicine," in *2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2023, pp. 1–9.

[11] L. Freund, "Towards a comprehensive theory of aligned emergence in ai systems: Navigating complexity towards coherence," *Journal of Artificial Intelligence Research*, 2023.

[12] A. Brem, F. Giones, and M. Werle, "The ai digital revolution in innovation: A conceptual framework of artificial intelligence technologies for the management of innovation," *IEEE Transactions on Engineering Management*, vol. 70, no. 2, pp. 770–776, 2021.

[13] D. Gibson, V. Kovanovic, D. Ifenthaler, S. Dexter, and S. Feng, "Learning theories for artificial intelligence promoting learning processes," *British Journal of Educational Technology*, vol. 54, no. 5, pp. 1125–1146, 2023.

[14] A. Haidar, "An integrative theoretical framework for responsible artificial intelligence," *International Journal of Digital Strategy, Governance, and Business Transformation (IJDSGBT)*, vol. 13, no. 1, pp. 1–23, 2024.

[15] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and kubernetes," *Communications of the ACM*, vol. 59, no. 5, pp. 50–57, 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2890784

[16] Prometheus, "Prometheus documentation," Available: https://prometheus.io/, accessed: October 1, 2024.

[17] InfluxData, "Influxdata documentation," Available: https://www.influxdata.com/index/, accessed: October 1, 2024.

[18] Grafana, "Grafana documentation," Available: https://grafana.com/, accessed: October 1, 2024.

[19] G. Labs, "Node exporter full dashboard for prometheus," https://grafana.com/grafana/dashboards/1860-node-exporter-full/, accessed: 2024-11-14.

[20] S. C. IO, "Kepler exporter grafana dashboard," https://github.com/sustainable-computing-io/kepler/blob/main/grafana-dashboards/Kepler-Exporter.json, accessed: 2024-11-14.

[21] M. Amaral, A. R. S. S. Oliveira, P. M. L. F. de Azevedo, and E. R. M. de Lima, "Kepler: A framework to calculate the energy consumption of containerized applications," in *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. Chicago, IL, USA: IEEE, 2023, pp. 69–71.

[22] MLFlow, "Mlflow documentation," Available: https://mlflow.org/, accessed: October 1, 2024.

[23] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[24] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001, available online: JSTOR.

[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[28] L. Pena, "Top 50 spotify songs - 2019," Available: https://www.kaggle.com/datasets/leonardopena/top50spotify2019, accessed: October 1, 2024.

[29] Scikit-learn, "Scikit-learn documentation," Available: https://scikit-learn.org/stable/, accessed: October 1, 2024.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2017, pp. 3149–3157. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 6638–6648. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/f5f8590cd58a54e94377e6ae2edba1a7-Paper.pdf

[32] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[33] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[35] M. Tan, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[36] A. G. Howard, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[40] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," *arXiv preprint arXiv:2305.09972*, 2023.

[41] A. Krizhevsky, G. Hinton *et al.*, *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, 2009.