scientific reports

Check for updates

OPEN

Comparative analysis of model compression techniques for achieving carbon efficient AI

Eileen Paula[1], Jayesh Soni[1]✉, Himanshu Upadhyay[2] & Leonel Lagos[3]

The growing computational demands of models, such as BERT, have raised concerns about their environmental impact. This study addresses the pressing need for sustainable Artificial Intelligence practices by investigating the efficiency of model compression techniques in reducing the energy consumption and carbon emissions of transformer-based models without compromising performance. Specifically, we applied pruning, knowledge distillation, and quantization to transformer-based models (BERT, DistilBERT, ALBERT, and ELECTRA) using the Amazon Polarity Dataset for sentiment analysis. We also compared the energy efficiency of these compressed models against inherently carbon-efficient transformer models, such as TinyBERT and MobileBERT. To evaluate each model's energy consumption and carbon emissions, we utilized the open-source tool CodeCarbon. Our findings indicate that applying model compression techniques resulted in a reduction in energy consumption of 32.097% for BERT with pruning and distillation, -6.709% for DistilBERT with pruning, 7.12% for ALBERT with quantization, and 23.934% for ELECTRA with pruning and distillation, while maintaining performance metrics within a range of 95.871–99.062% accuracy, precision, recall, F1 score, and ROC AUC except for ALBERT with quantization. Specifically, BERT with pruning and distillation achieved 95.90% accuracy, 95.90% precision, 95.90% recall, 95.90% F1-score, and 98.87% ROC AUC; DistilBERT with pruning achieved 95.87% accuracy, 95.87% precision, 95.87% recall, 95.87% F1-score, and 99.06% ROC AUC; ELECTRA with pruning and distillation achieved 95.92% accuracy, 95.92% precision, 95.92% recall, 95.92% F1-score, and 99.30% ROC AUC; and ALBERT with quantization achieved 65.44% accuracy, 67.82% precision, 65.44% recall, 63.46% F1-score, and 72.31% ROC AUC, indicating significant performance degradation due to quantization sensitivity in its already compressed architecture. Overall, this demonstrates the potential for sustainable Artificial Intelligence practices using model compression.

Keywords Energy-efficient AI, Model compression, NLP model sustainability

Abbreviations

AI	Artificial Intelligence
ALBERT	A lite BERT for self-supervised learning of language representations
BERT	Bidirectional encoder representations from transformers
BiLSTM	bidirectional long short-term memory
CI	Confidence interval
CNN	Convolutional neural network
CPU	Central processing unit
ELECTRA	Efficiently learning an encoder that classifies token replacements accurately
GPU	Graphics processing unit
GRU	Gated recurrent unit
KL	Kullback–Leibler
L1	Pruning L1 unstructured pruning
LSTM	Long short-term memory
MAE	Mean absolute error
MCA	Multi-channel analyzer
MSE	Mean squared error

[1]Applied Research Center, Florida International University, Miami 33174, USA. [2]Department of Electrical and Computer Engineering, Florida International University, Miami 33174, USA. [3]Moss Department of Construction Management, Florida International University, Miami 33174, USA. ✉email: jsoni@fiu.edu

NLP	Natural language processing
SD	Standard deviation
PTQ	Post-training quantization
RAM	Random access memory
ReLU	Rectified linear unit
RNN	Recurrent neural network
RMSE	Root mean squared error
ROC AUC	Receiver operating characteristic—area under the curve
R^2	Coefficient of determination
TDP	Thermal design power
T4 GPU	NVIDIA Tesla T4 graphics processing unit

Artificial Intelligence (AI) is advancing rapidly, contributing to the increasing demand for substantial computational resources in deep learning models, resulting in large amounts of energy being consumed. Large-scale models, such as Bidirectional Encoder Representations from Transformers (BERT), have achieved high performance in natural language processing (NLP) tasks. However, this performance comes with significant environmental costs due to its high energy demands. Recent studies have highlighted the substantial carbon footprint associated with training large AI models. For instance, researchers estimated that training a single large language model could emit approximately 300,000 kg of carbon dioxide, an amount comparable to 125 round-trip flights between New York and Beijing[1]. This underscores the pressing need for energy-efficient AI development, as the environmental cost of scaling deep learning models continues to rise.

As a result, reducing AI models' energy consumption and carbon emissions has become a critical research area, proven by the increase in publications in this area, as seen in Fig. 1. To ensure reproducibility, we searched Dimensions.ai for "AI Energy Consumption," "AI Carbon Emissions," and "AI Model Compression" separately, applying the Publication Year filter to record annual publication counts from 1989 to 2024. Model compression techniques, including pruning, knowledge distillation, and quantization, have been explored to mitigate this issue[2–4]. Pruning is the process of reducing the number of parameters in a model by removing components, such as weights or neurons, all while maintaining its performance[5]. Knowledge distillation is transferring knowledge from a large model to a smaller one. The larger model is often referred to as the teacher model and the smaller model is often referred to as the student model. In order to learn from the teacher, the student approximates the teacher's output predictions, typically using a technique like matching the teacher's logits or soft labels[6]. Quantization is the process of reducing the numerical precision of the model's parameters from their original floating-point representation to a lower bit-width format which is done using a scale factor and mapping them to discrete levels[7]. Recent studies have investigated the combinations of these compression techniques. For example, Malihi and Heidemann[3] introduced a framework combining knowledge distillation and pruning to achieve efficient model compression. Their approach applies knowledge distillation followed by pruning and fine-tuning, enabling significant reductions in model size while maintaining the model's performance. Similarly, techniques like Early Pruning with Self-Distillation (EPSD) have been proposed to integrate pruning and distillation effectively[8].

Despite these advances, there remains a gap in understanding the practical implications of applying these compression techniques to models in tasks often seen in the real world. While previous work has focused on model performance metrics, few studies have quantified the environmental benefits of compression techniques

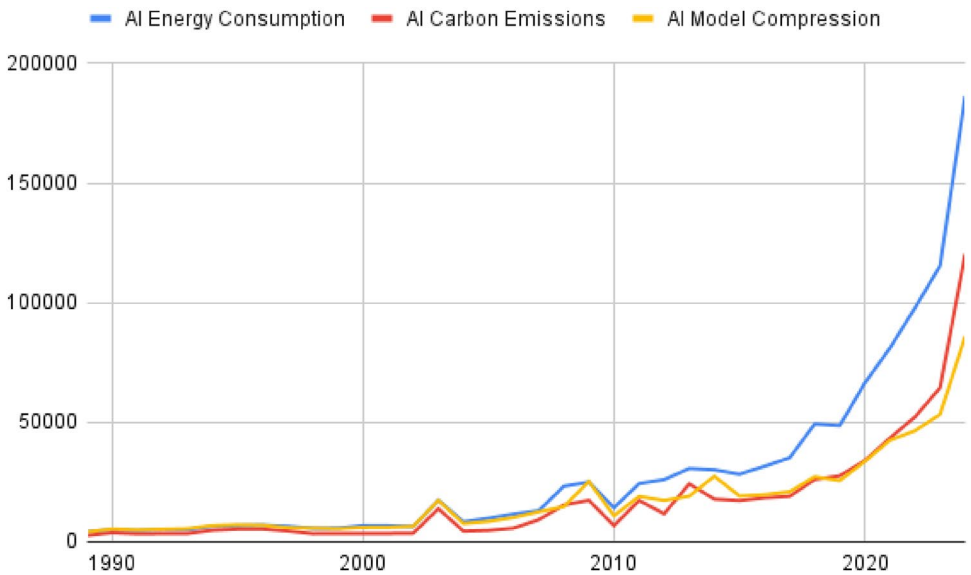

Fig. 1. The number of papers on AI energy consumption, AI carbon emissions, and AI model compression from 1989 to 2024. Data collected from https://app.dimensions.ai/.

when applied to large-scale datasets. In order to measure the energy consumption and carbon emissions associated with AI models, tools like CodeCarbon[9] and CarbonTracker[10] have been developed. These tools allow users to monitor energy consumption and carbon emissions during model training and inference, facilitating more sustainable AI practices. Tracking the energy footprint is crucial for understanding the trade-offs between model performance and environmental impact.

Research motivation

Sentiment analysis is a fundamental task in NLP with notable commercial and social relevance. The Amazon Polarity dataset serves as an ideal benchmark for studying energy-efficient AI models in real-world applications. This dataset reflects practical scenarios where AI models are deployed in consumer-facing applications requiring efficient processing and analyzing large quantities of unstructured text data. Consequently, energy consumption becomes a concern. By focusing on the Amazon Polarity dataset, we can evaluate the effectiveness of model compression techniques in handling substantial amounts of text data while minimizing computational resources. Studying energy-efficient models on such datasets allows us to directly assess potential energy usage and carbon emissions reductions when applying compression techniques. As the demand for AI-driven services grows, so does the environmental impact of training and deploying large models. There is a pressing need to develop methods that can reduce the carbon footprint of AI without sacrificing accuracy or utility. By demonstrating that compressed models can maintain high performance on significant tasks like sentiment analysis, we contribute to the broader goal of sustainable AI development. This approach aligns with recent initiatives aiming to reduce the environmental impact of AI technologies[11].

Key contribution

In this study, we aim to address large AI models' environmental impact by exploring model compression techniques' effectiveness in reducing energy consumption and carbon emissions. Our key contributions are as follows:

1. Comprehensive evaluation of compression techniques on transformer-based models: We systematically apply pruning, knowledge distillation, and quantization to transformer-based models, including BERT, DistilBERT[12], ALBERT[13], and ELECTRA[14]. While previous studies have explored compression techniques, our work focuses on their application to a variety of transformer architectures in the context of a large-scale sentiment analysis task. By implementing these compression methods, we reduce model sizes and computational demands, assessing their effects on both performance and energy efficiency.
2. Quantitative analysis of energy consumption and carbon emissions: We utilize CodeCarbon[9] to monitor and quantify the energy consumption and estimated carbon emissions during model training and inference. This analysis offers insights into the relationship between model performance and environmental impact, a relationship that has not been explored enough in prior research.
3. Guidelines for sustainable AI model deployment: We extensively evaluate the compressed models and compare performance metrics with energy consumption metrics. Our findings provide practical guidelines for selecting and deploying energy-efficient AI models in real-world applications, therefore contributing to the advancement of sustainable AI practices. Additionally, we compare compressed models to TinyBERT and MobileBERT, which are pretrained lightweight transformers designed for efficiency. This comparison strengthens the evaluation by providing a baseline for understanding whether compression techniques can outperform existing carbon-efficient architectures.

By addressing this issue, our research fills the gap in understanding the practical implications of model compression on energy consumption and carbon emissions in transformer-based models. We demonstrate that compression techniques can be used to significantly reduce energy consumption and carbon emissions, aligning AI development with environmental sustainability goals. We maintain a repository on https://github.com/eileenpaula/Achieving-Carbon-Efficient-AI that contains the code used to conduct this experiment.

Structure

This paper is structured as follows: The next section provides a Literature Review on model compression techniques and energy-efficient AI, focusing on pruning, knowledge distillation, and quantization. The Methods section outlines the dataset, models, compression techniques, and carbon measurement tools used. The Experimental Setup describes the hardware, training configurations, and model-compression combinations. The Results section presents performance metrics, energy consumption trends, and trade-offs. The Discussion interprets these findings, emphasizing their implications for sustainable AI. The Future Work section suggests potential extensions, including task-specific energy optimizations and broader dataset evaluations. Finally, the Conclusion summarizes key takeaways and their relevance to environmentally sustainable AI.

Literature review

The increasing demand for energy-efficient AI models has led to extensive research on optimizing energy consumption in both AI and renewable energy systems. This section reviews recent advancements in this domain, focusing on model compression techniques and their environmental impact. While significant progress has been made in improving computational efficiency, there remains a gap in research that systematically quantifies the environmental impact of these techniques, particularly their effect on carbon emissions during training and inference phases. This study aims to address this gap by bridging the divide between theoretical advancements in model compression and their practical implications for AI sustainability.

One study explored various Deep Neural Network (DNN) topologies, including Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network-LSTM (CNN-LSTM), for renewable energy forecasting[15]. The researchers demonstrated that tuned LSTM architectures significantly outperformed traditional models, achieving a Mean Absolute Error (MAE) of 0.08765, Mean Squared Error (MSE) of 0.00876, Root Mean Squared Error (RMSE) of 0.09363, and an R-squared (R^2) value of 0.99234. The incorporation of meteorological variables, temporal variables, and hyperparameter tuning, was critical in achieving this performance. This study highlights the role of advanced DNNs in promoting environmentally sustainable electricity systems through improved forecasting accuracy.

In addition to individual applications, recent studies have expanded the scope of energy-efficient AI to include environmental monitoring, combustion systems, and renewable forecasting. One set of findings highlights that incorporating meteorological and temporal features into optimized LSTM and BiLSTM models can significantly improve forecasting accuracy and reduce computational inefficiency[16]. Another study proposed a hybrid physics-informed deep learning approach for predicting PM2.5 concentrations using fully convolutional architectures, reducing the need for handcrafted features and enhancing inference efficiency[17]. A related work applied a variety of machine learning and deep learning models to predict the temperature of cryogenic fuel injections in combustion systems, with a coupled fully connected neural network and Extra Trees Regressor achieving near-perfect predictive performance[18]. Finally, the performance of several DNN architectures was benchmarked for renewable energy forecasting tasks, showing that fine-tuned LSTM variants achieved exceptional accuracy when trained with enriched feature sets and optimized hyperparameters[15]. Together, these studies reinforce the growing trend of integrating structural optimization, hybrid modeling, and domain knowledge to advance the energy efficiency and predictive power of AI systems across diverse applications.

A similar focus on AI driven energy efficiency is seen in research on optimizing smart home energy management using machine learning techniques[19]. This study investigated predictive models that enhance energy efficiency while ensuring user comfort, leveraging reinforcement learning algorithms to dynamically adjust energy usage patterns. The proposed AI model demonstrated improved predictive performance, achieving an MSE of 0.02284, MAE of 0.184, and RMSE of 0.15113. These findings reinforce the role of AI in sustainable energy management, where efficient modeling translates to direct energy savings in real-world applications.

In addition to individual applications, there has been an increasing focus on Green AI, an initiative that promotes energy-efficient AI development. One study introduced a comprehensive framework for sustainable AI, emphasizing model optimization techniques such as pruning, quantization, and knowledge distillation[20]. Their findings show that pruning can reduce the number of parameters in neural networks by up to 90% while maintaining competitive accuracy. For instance, in AlexNet, reducing parameters from 61 million to 6.7 million led to a 90% reduction in energy consumption with only a minor 0.5% accuracy drop. These findings suggest that compression techniques can significantly reduce computational overhead without compromising performance.

Similarly, research on the environmental costs of training large-scale models has emphasized the role of hardware in mitigating AI's carbon footprint[21]. One study demonstrated that switching from an NVIDIA T4 GPU to an A100 GPU resulted in an 83% average reduction in CO2 emissions, with some configurations achieving reductions as high as 99.7%, all while maintaining model performance. This highlights the importance of both model compression and hardware efficiency in reducing AI's environmental impact.

In addition to compression techniques, several other studies have proposed lightweight transformer architectures that achieve efficiency gains without explicit compression. TinyBERT[22] is a 4 layer student model trained with knowledge distillation, achieving a 7.5 times smaller model size and a 9.4 times faster speedup over BERT while retaining 96.8% of its accuracy on GLUE. MobileBERT[23] employs an inverted-bottleneck structure and achieves a 4.3 times larger reduction in model size while maintaining competitive performance on NLP benchmarks. We compare these architectures against traditional compression techniques to assess their relative efficiency in energy consumption.

While model compression techniques have predominantly been studied in the context of NLP tasks, recent research has explored their implications in other domains. For example, the Green AI framework[20] not only applies to NLP models but also discusses energy-efficient deep learning in computer vision and speech recognition. Studies on pruning and quantization for convolutional neural networks (CNNs) have shown that these methods can significantly reduce the energy consumption of image classification models while maintaining high performance[24,25]. These findings reinforce the applicability of compression techniques in making AI more sustainable. Another area where energy-efficient AI is gaining traction is real-time AI applications. AI-driven smart home energy management[19] and renewable energy forecasting[15] demonstrate that compression techniques can enable more efficient AI deployment in resource-constrained environments. While previous research has primarily focused on improving deep learning model accuracy and computational efficiency, fewer studies have analyzed their direct impact on energy consumption at scale. This study aims to fill that gap by systematically analyzing model compression's trade-offs between carbon emissions and computational efficiency across different AI applications.

Despite these advancements, many studies either focus on theoretical improvements in compression or provide high-level estimates of carbon emissions without detailed empirical analysis. This study distinguishes itself by systematically quantifying the environmental impact of model compression during both training and inference. Unlike prior research that primarily optimizes for computational efficiency or accuracy retention, our work provides direct measurements of energy consumption and CO2 emissions when applying pruning, knowledge distillation, and quantization to transformer-based models. By integrating compression techniques with energy tracking tools such as CodeCarbon, this study presents a data-driven analysis of how compression affects energy efficiency at a granular level. Instead of focusing solely on theoretical energy reductions, our approach provides real-world emissions data, making it more applicable to industry and environmental policy

discussions. Through this evaluation, we contribute to the broader goal of developing sustainable AI frameworks that balance performance with environmental responsibility.

Methods

Dataset

The Amazon Polarity Dataset is designed for sentiment polarity analysis and was part of the Stanford Network Analysis Project. This dataset consists of product reviews collected over 18 years, containing 34,686,770 reviews from 6,643,669 users on 2,441,053 products[26]. The dataset captures user sentiment through polarity labeling, classifying reviews as either positive or negative, based on the user's star rating. Reviews with scores of 1 and 2 are labeled as negative, 4 and 5 are labeled as positive, and a score of 3 is excluded to maintain binary sentiment polarity.

Data analysis was done on the dataset to find the distribution of reviews. Our analysis showed that each class in the dataset contains 1,800,000 reviews for training and 200,000 for testing. This large scale makes it well-suited for building sentiment analysis models capable of generalizing across a broad set of reviews. However, due to time and computational constraints, only a 75% subset of the original training data was used in this study. This subset includes approximately 2,700,000 reviews, split evenly between positive and negative classes, providing a balanced representation while reducing training time and computational requirements.

The distribution of labels in the training subset shows an even split, with 50% of the reviews labeled as positive and 50% as negative, as illustrated in Fig. 2. Additionally, the comparison between the training subset and the full test set, shown in Fig. 3, highlights that the training subset constitutes 87.1% of the data used, while the test set comprises 12.9%. This balanced and structured approach enables effective evaluation of model performance and energy consumption on sentiment classification tasks.

Table 1 shows the key facts of the Amazon Polarity Dataset dataset. The dataset's focus on sentiment polarity makes it particularly valuable for sentiment analysis tasks in NLP, enabling the evaluation of models' capabilities to classify text-based user feedback accurately. This relevance to practical applications, such as e-commerce and customer satisfaction analysis, highlights its significance in developing environmentally conscious, energy-efficient AI models without compromising accuracy[26]. The Amazon Polarity Dataset was selected over other benchmark sentiment analysis datasets due to its scale, structured sentiment labeling, and commercial relevance. Unlike smaller datasets such as IMDB (50K reviews) and SST-2 (67K phrases), which are domain-specific and limited in scope, Amazon Polarity provides a large-scale, product-focused corpus, making it more representative of real-world applications in business intelligence, automated review analysis, and large-scale sentiment classification. It is also well-suited for evaluating energy-efficient AI models, as its balanced class distribution and high-volume text data create a computationally demanding yet controlled benchmark for comparing baseline and compressed models. Additionally, when compared to datasets such as Yelp Reviews (6M reviews) and Twitter Sentiment Analysis (1.6M tweets), Amazon Polarity's structured binary sentiment format ensures a more standardized evaluation, reducing variability caused by nuanced or hierarchical sentiment labels. Due to time constraints, only 75% of the dataset was used for training. This reduction in training data significantly decreased the time required for model fine-tuning while still maintaining a representative subset that preserves the dataset's balance and distribution. This approach allowed for a more efficient evaluation of model compression techniques without sacrificing the validity of energy consumption and performance comparisons. By maintaining a large yet computationally feasible dataset, the study ensures that findings remain applicable to real-world NLP deployments where scalability and efficiency are critical considerations.

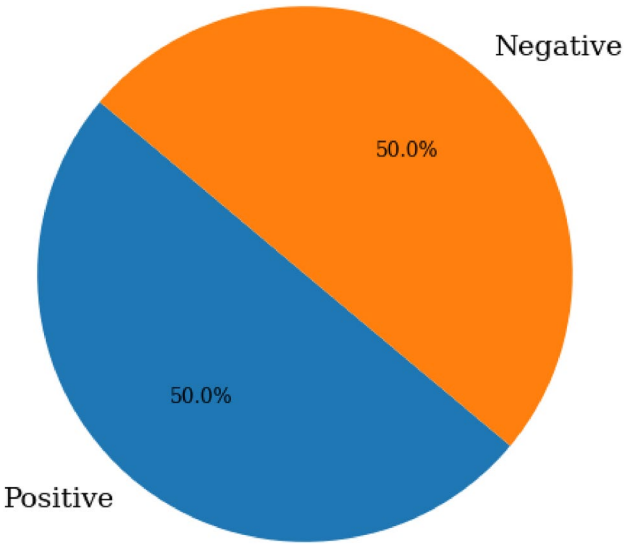

Fig. 2. Visual representation of label distribution in the training subset of the Amazon Polarity Dataset.

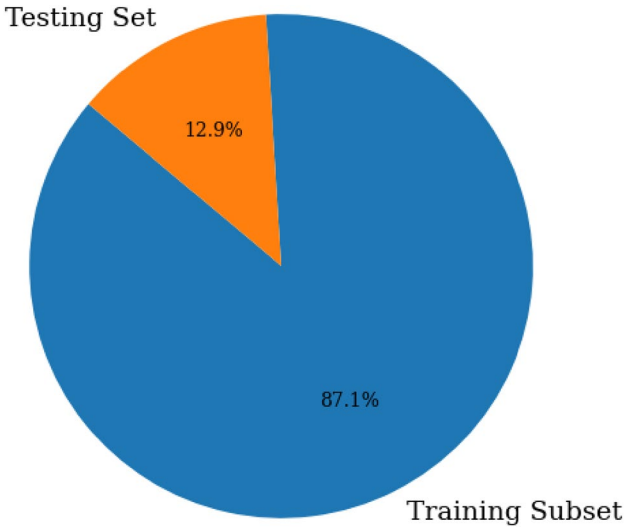

Fig. 3. Visual representation of proportion of training subset to the testing set in the Amazon Polarity Dataset.

Data span	18 years
Total reviews (full dataset)	3,600,000 reviews
Training set size (subset)	2,700,000 reviews
Test set size (full)	400,000 reviews
Languages	Primarily English
Task	Sentiment classification (positive/negative)

Table 1. Key facts about the Amazon Polarity Dataset used in this study, summarizing the dataset's scope, subset size, language, and relevance to sentiment classification tasks.

Models

In this study, we trained four transformer-based models for sentiment analysis: BERT, DistilBERT, ALBERT, and Electra, alongside two pretrained carbon efficient transformer models, TinyBERT and MobileBERT. This was conducted using a total of three trials. Each model was selected based on its unique architecture and efficiency characteristics, allowing for a comparative performance analysis under various compression techniques. TinyBERT and MobileBERT serve as additional baselines, representing transformer architectures that are designed from the outset to be computationally efficient, providing insight into whether targeted compression techniques offer additional benefits over purpose-built lightweight models.

1. BERT: BERT has become a benchmark in natural language processing due to its bidirectional transformer architecture, enabling it to capture complex context by looking at both past and future tokens in a sentence[27]. However, its architecture is computationally intensive, leading to higher energy consumption. The multi-head attention mechanism is central to BERT's computational requirements, allowing it to simultaneously attend to different representation subspaces. Multi-head attention is formulated as follows:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O, \quad \text{where} \quad \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{1}$$

Additionally, the self-attention mechanism used by each attention head is defined as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where Q, K, and V represent the query, key, and value matrices, respectively. These equations illustrate the quadratic complexity of self-attention, where computational cost scales with sequence length and the number of attention heads[28]. This means that as input sequences grow longer, BERT requires significantly more memory and compute power, leading to high energy consumption and carbon emissions. This computational intensity underscores why model compression techniques such as pruning and distillation are necessary to reduce BERT's resource footprint while maintaining performance.

Figure 4 shows the BERT architecture, which is composed of multiple stacked Transformer encoder layers using multi-head self-attention. BERT is known for its bidirectional attention mechanism, which allows it to capture rich contextual relationships in language. This has made it a foundational model for numerous NLP tasks due to its strong performance. However, BERT's main limitation lies in its computational demands. Its large number of parameters and quadratic attention complexity with respect to sequence length makes it memory and energy intensive, especially during pretraining and fine-tuning. This motivates the exploration of model compression techniques to reduce its resource footprint without sacrificing performance.

2. DistilBERT: DistilBERT, a distilled version of BERT, reduces the number of parameters by roughly 40% while retaining 97% of BERT's language understanding capabilities[12]. This model was trained using knowledge distillation, where a smaller model (student) learns to replicate the behavior of a larger model (teacher), in this case, BERT. The distillation process is guided by minimizing a weighted combination of losses, as shown by the following equation:

$$L = (1 - \alpha) \cdot L_{CE}(p_s(1), y) + \alpha \cdot L_{KL}(p_s(\tau), p_t(\tau)), \tag{3}$$

where L_{CE} is the cross-entropy loss with "hard" labels, L_{KL} represents the Kullback-Leibler divergence between the student and teacher outputs, and τ is a temperature-scaling parameter that softens the teacher's output probabilities, providing additional information to the student model. Here, α controls the trade-off between the two losses, with L_{KL} playing a crucial role in aligning the student's predictions to mimic the teacher's output distribution[29]. The inclusion of the Kullback-Leibler divergence loss allows the student model to capture subtle relationships between tokens that may be lost in traditional hard-label training. By leveraging softened teacher outputs, DistilBERT reduces computational complexity while preserving key linguistic patterns from BERT. Additionally, the removal of redundant layers leads to faster inference and lower energy consumption, making DistilBERT a strong candidate for further compression using pruning techniques. This study explores whether additional pruning can further optimize its efficiency while maintaining competitive performance.

DistilBERT was chosen for its balance between computational efficiency and performance, making it ideal for applications where maintaining high accuracy is essential under limited resources. This study applies further pruning techniques to DistilBERT to explore whether additional energy savings can be achieved while retaining competitive accuracy. By leveraging the combination of distillation and pruning, this research aims to quantify the effectiveness of these strategies for energy-efficient NLP model deployment.

Figure 5 illustrates the DistilBERT architecture, which removes every other layer from BERT and simplifies the architecture while preserving core attention mechanisms. Its main advantage is computational efficiency: it retains 97% of BERT's performance while being 40% smaller and 60% faster. However, DistilBERT uses a single-phase task-agnostic distillation, which may limit its adaptation to specialized downstream tasks compared to models that also include task-specific distillation like TinyBERT.

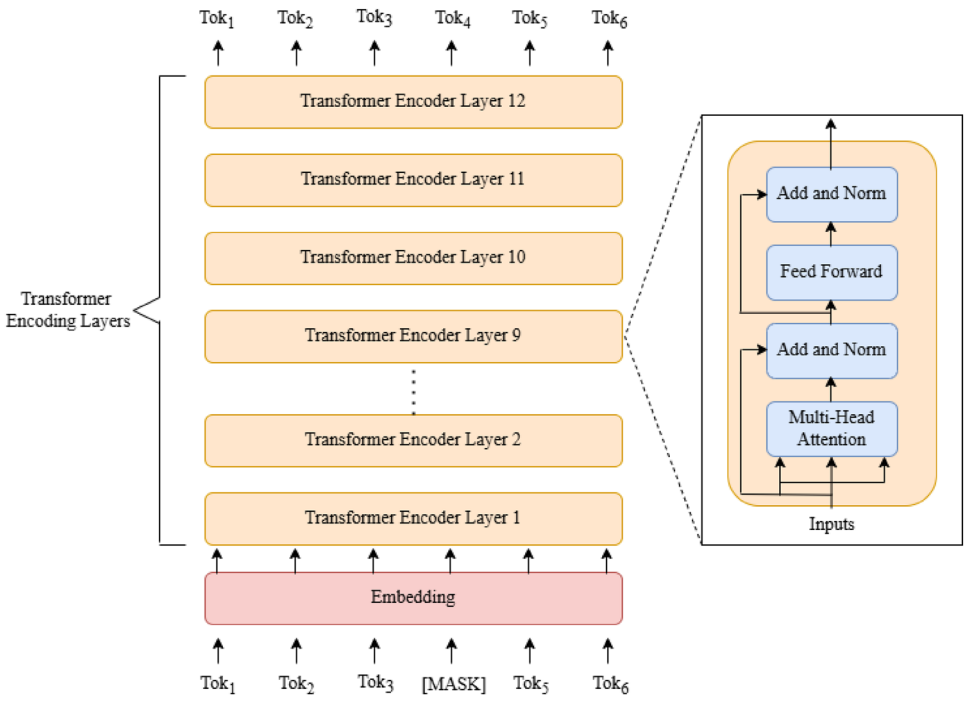

Fig. 4. BERT architecture with 12 transformer encoder layers and bidirectional attention.

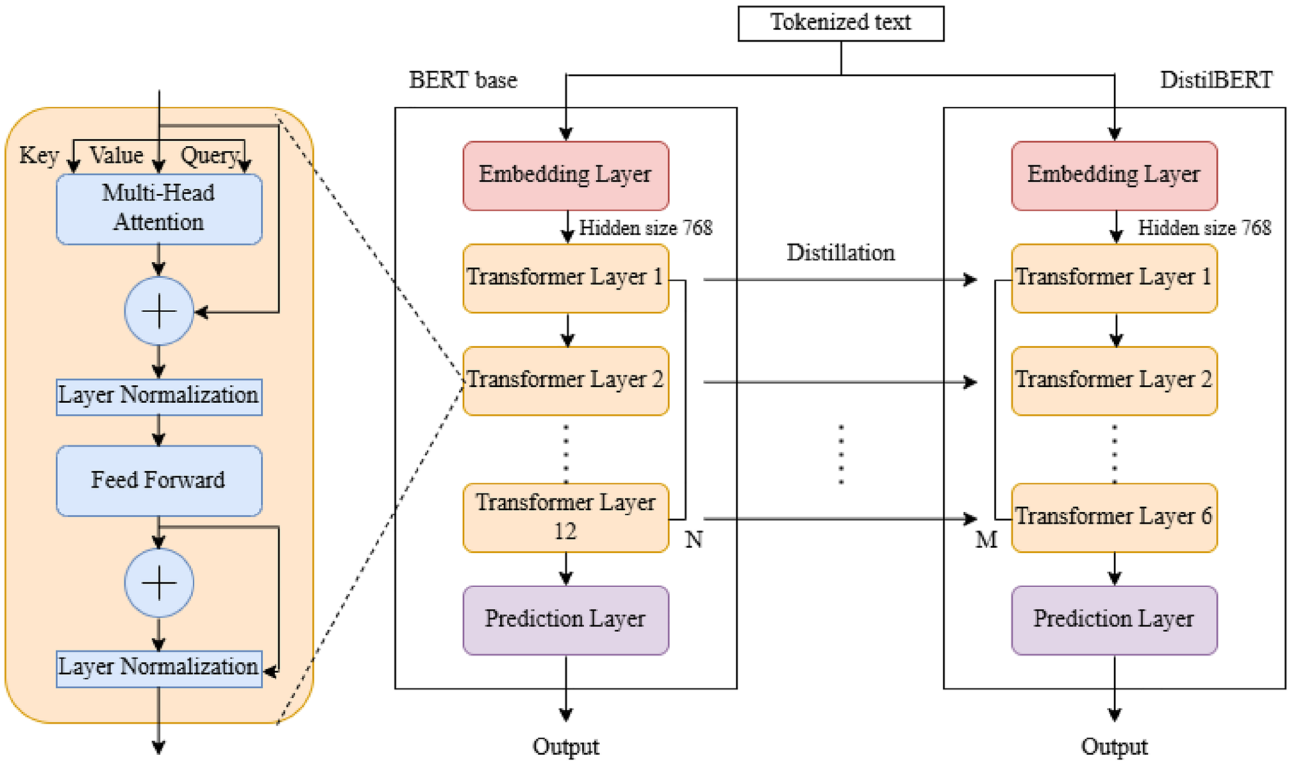

Fig. 5. DistilBERT architecture showing reduced depth and distillation from BERT.

3. ALBERT: ALBERT was designed to address the limitations of BERT by reducing the model size and increasing efficiency, primarily through parameter-sharing and factorized embedding parameterization[13]. Its architecture drastically reduces memory usage and allows for faster training without a significant trade-off in performance. ALBERT achieves this efficiency by introducing factorized embedding parameterization, where the total parameters are defined as:

$$\text{Total parameters} = O(V \times E + E \times H) \tag{4}$$

here, V is the vocabulary size, E is the embedding size, and H is the hidden layer size. This approach enables ALBERT to scale the hidden size H without significantly increasing embedding parameters $O(V \times H)$. Additionally, ALBERT employs cross-layer parameter sharing, which involves sharing weights across layers to reduce model size further while stabilizing training. This equation highlights how ALBERT reduces memory usage and computational cost by factorizing embeddings and reusing parameters across layers. These modifications significantly lower model size and training memory requirements, making ALBERT a computationally efficient alternative to BERT. ALBERT was selected in this study to explore the effects of quantization on an inherently compact model and to analyze whether additional compression can enhance its efficiency further, particularly in reducing energy consumption during training and inference. As shown in Fig. 6, ALBERT modifies the standard BERT architecture by factorizing the embedding layer and applying cross-layer parameter sharing. Its advantages include reduced parameter count and memory footprint while retaining performance on many NLP benchmarks. These optimizations make ALBERT scalable and efficient. A key trade-off is that parameter sharing across layers may reduce the model's ability to capture hierarchical features, limiting representational diversity.

4. Electra: Electra is designed to be efficient in both model size and training approach. Unlike BERT's masked language model training, ELECTRA adopts a novel pretraining strategy called replaced token detection. In this approach, the generator predicts masked tokens in a given sequence, forming a corrupted sequence, which the discriminator then evaluates to distinguish between real and replaced tokens[14]. The generator in ELECTRA performs masked language modeling (MLM), predicting the identities of masked tokens in the input. The masked language modeling loss for the generator, L_{MLM}, is formulated as:

$$L_{\text{MLM}}(x, \theta_G) = \mathbb{E}\left[\sum_{i \in m} -\log p_G(x_i | x_{\text{masked}})\right] \tag{5}$$

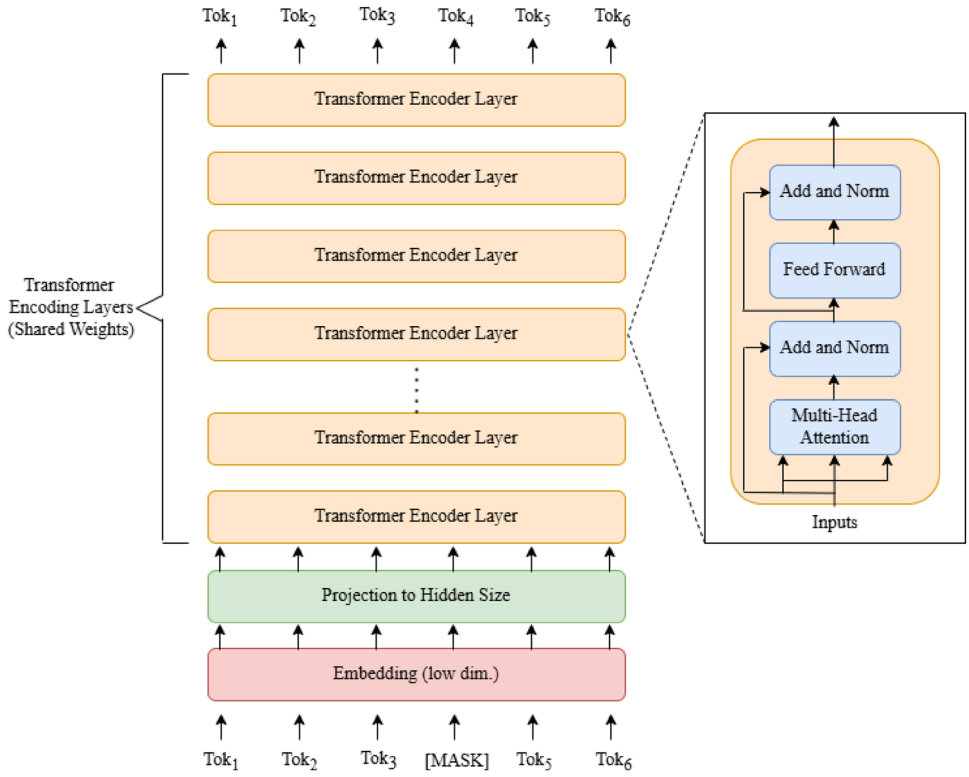

Fig. 6. ALBERT architecture with cross-layer parameter sharing and factorized embeddings.

where $p_G(x_i|x_{\text{masked}})$ is the probability that the generator assigns to the true token x_i given the masked sequence x_{masked}, and θ_G represents the parameters of the generator. The discriminator is then trained to detect which tokens in the corrupted sequence x_{corrupt} have been replaced by generator predictions rather than being part of the original input. The discriminator's objective is formalized as:

$$L_{\text{Disc}}(x, \theta_D) = \mathbb{E}\left[\sum_{t=1}^{n} -\mathbb{1}(x^t_{\text{corrupt}} = x^t) \log D(x_{\text{corrupt}}, t) - \mathbb{1}(x^t_{\text{corrupt}} \neq x^t) \log(1 - D(x_{\text{corrupt}}, t))\right] \quad (6)$$

where $D(x_{\text{corrupt}}, t)$ represents the probability that the discriminator assigns to token t in x_{corrupt} being "real" (matching the original). Unlike BERT, which learns by predicting masked tokens, ELECTRA's Replaced Token Detection (RTD) pretraining method trains the model to distinguish real tokens from generated ones, making it a more computationally efficient alternative to masked language modeling. This approach allows ELECTRA to learn representations more effectively while reducing redundant computations, which in turn lowers training resource requirements. This study applies pruning and distillation techniques to ELECTRA, aiming to evaluate how these additional optimizations impact computational efficiency and energy consumption. Figure 7 visualizes ELECTRA's architecture with a generator-discriminator setup trained via replaced token detection. ELECTRA is significantly more sample-efficient than BERT, as it learns from all input tokens rather than only those that are masked. This results in faster convergence and more efficient use of compute resources. However, its architecture involves coordinating two separate networks (generator and discriminator), increasing implementation complexity. Also, the pretraining task is not identical to downstream objectives, which may require careful fine-tuning.

5. TinyBERT: TinyBERT is a lightweight version of BERT designed to improve computational efficiency while retaining high accuracy[22]. Unlike larger transformer models that require additional compression techniques, TinyBERT is trained through a two-stage knowledge distillation process, where both pretraining and task-specific fine-tuning knowledge are transferred from a larger teacher model. This approach allows TinyBERT to achieve a 7.5 times smaller model size and 9.4 times faster inference speed than BERT while maintaining 96.8% of BERT's performance on GLUE benchmarks. TinyBERT's compact design reduces the number of transformer layers while preserving key self-attention mechanisms, making it well-suited for real-time NLP applications. Due to its efficiency and lower computational footprint, TinyBERT serves as an important baseline in this study to determine whether explicitly applying pruning, distillation, and quantization to larger models provides additional energy savings beyond what TinyBERT already achieves. Figure 8 shows the TinyBERT architecture, which reduces depth while maintaining the attention structure of BERT.

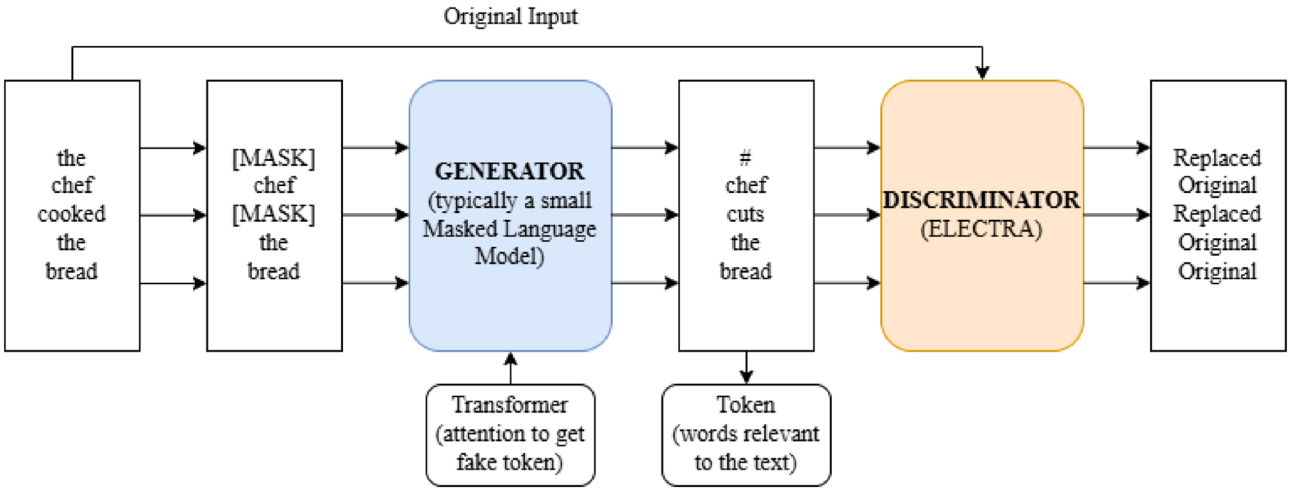

Fig. 7. ELECTRA architecture with generator-discriminator pretraining for replaced token detection.

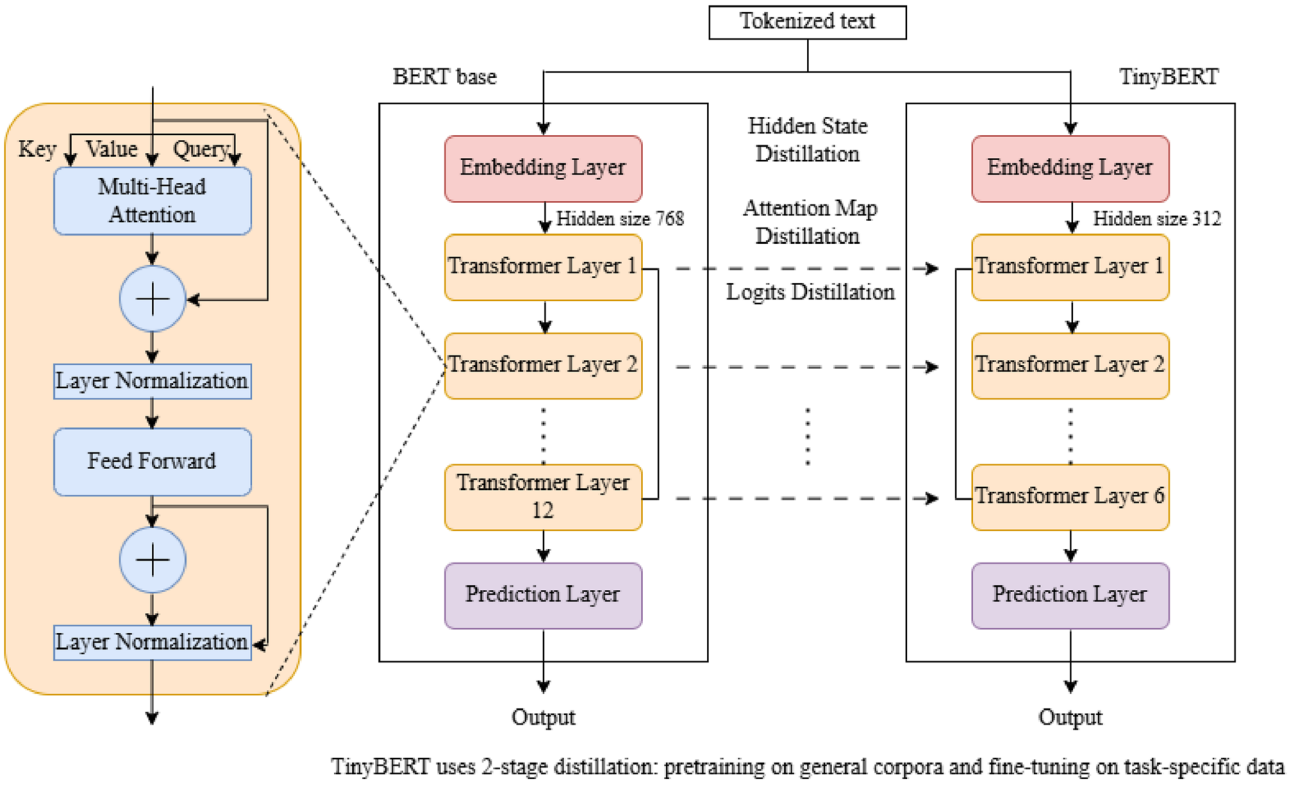

Fig. 8. TinyBERT architecture with a compact encoder trained via two-stage distillation.

An advantage of TinyBERT is that it delivers strong performance with a significantly reduced model size, achieving rapid inference and lower energy consumption. It is trained using a two-stage distillation framework: task-agnostic pretraining distillation and task-specific fine-tuning distillation. A limitation of TinyBERT is that its reduced size can limit its representation power on complex tasks, especially those requiring deeper contextual modeling.

6. MobileBERT: MobileBERT is designed to deliver high-performance NLP capabilities while being optimized for low-resource devices such as mobile phones and edge computing[23]. MobileBERT retains the depth of the original BERT model but applies a bottleneck structure to reduce computational complexity while maintaining expressiveness. MobileBERT achieves 4.3 times smaller model size, 5.5 times faster inference speed, and 75% fewer parameters than BERT. These improvements make MobileBERT a strong alternative to compression-based approaches for achieving energy efficiency in transformer-based models. Similar to TinyBERT,

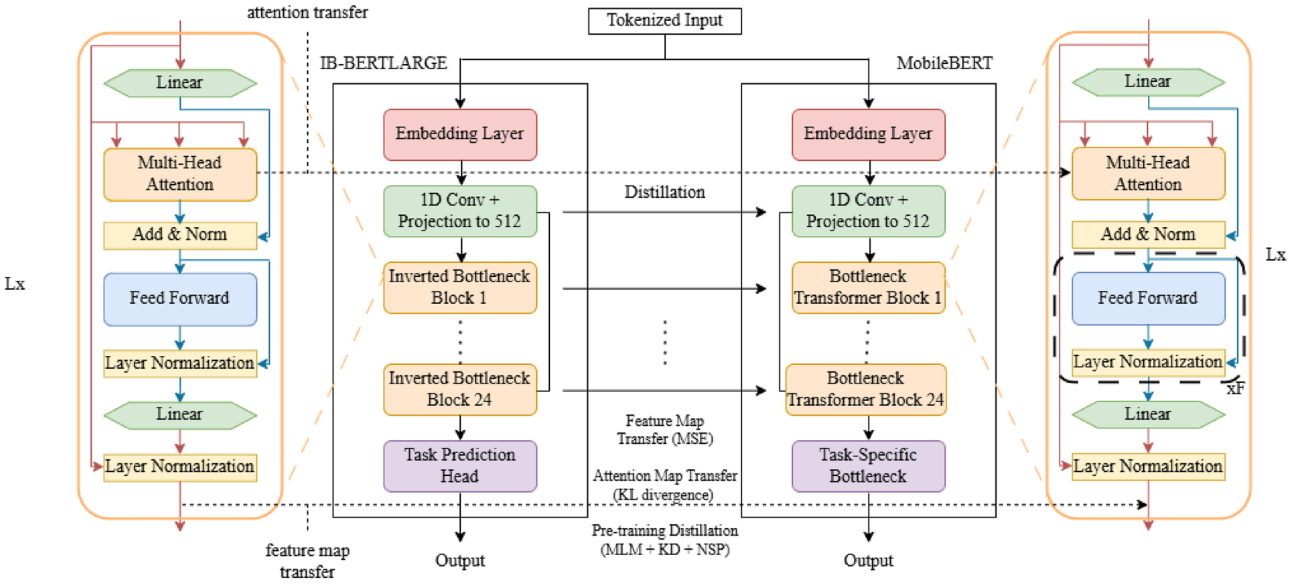

Fig. 9. MobileBERT architecture with bottleneck layers and task-specific distillation.

```
def apply_pruning(model, amount=0.1):
    for name, module in model.named_modules():
        if isinstance(module, torch.nn.Linear):
            prune.l1_unstructured(module, name='weight', amount=amount)
            prune.remove(module, 'weight')  # Permanently apply pruning
```

MobileBERT serves as a baseline to determine whether explicitly applying compression techniques to larger models provides additional energy savings in comparison to what MobileBERT currently accomplishes. The MobileBERT architecture is depicted in Fig. 9, highlighting its use of a bottleneck structure with a linear projection layer before the transformer encoder. MobileBERT retains the full depth of BERT but introduces a bottleneck design along with a projection layer to reduce width and computation cost, enabling it to perform efficiently on mobile and edge devices. It uses a teacher–student framework for distillation, learning from an inverted-bottleneck teacher model. Its increased architectural complexity and reliance on extensive pretraining and distillation make it less flexible for extension or modification compared to simpler models.

These models were selected to represent a range of transformer architectures with varying levels of computational complexity and energy requirements. Their diversity in design and compression techniques allows for a comprehensive comparison of how model architecture impacts the effectiveness of pruning, knowledge distillation, and quantization methods in reducing energy consumption without compromising model performance. Each model's unique structure offers insights into the trade-offs between model size, accuracy, and energy efficiency, enabling a well-rounded evaluation of sustainable AI techniques. In addition to evaluating these compression techniques, we also evaluate TinyBERT and MobileBERT to determine whether optimized architectures can match or outperform traditional transformers with compression applied. This comparison helps to identify whether targeted compression techniques offer further energy savings beyond what is already achieved by compact transformers.

Compression techniques

We implemented three primary compression techniques to reduce the computational resources required for large-scale transformer models: pruning, knowledge distillation, and quantization. We used L1 Unstructured Pruning, Knowledge Distillation, and Post-Training Quantization. Each technique was selected for its ability to reduce model size while maintaining performance.

1. Pruning: Pruning is a compression technique that reduces the model size by removing redundant or low-magnitude weights in the neural network, thereby reducing the model's computational requirements and memory footprint. In this study, we applied L1 Unstructured Pruning, a method that removes individual neural network weights based on their L1 norm (absolute value), minimizing model loss while enforcing a sparsity constraint. Figure 10 presents a visual representation of this type of pruning. Unlike structured pruning, it operates independently on each weight, identifying the least important weights and setting them to zero to create sparsity without altering the network's structure. This approach aims to minimize loss across all training samples while limiting the number of non-zero weights, ensuring a balance between performance and efficiency[5]. L1 Unstructured Pruning was applied to the linear layers of transformer models DistilBERT, BERT, and ELECTRA in this experiment. The pruning removed 10% of the least important weights

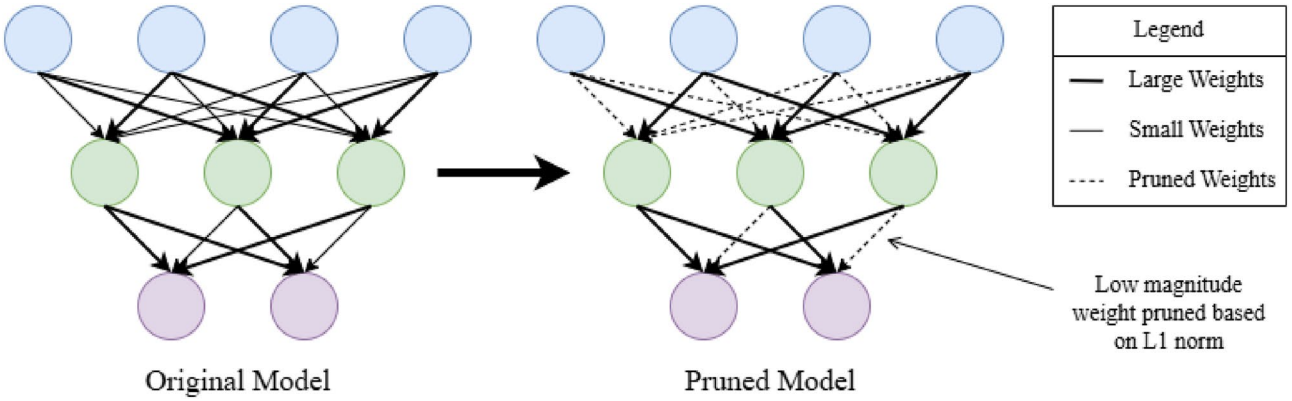

Fig. 10. L1 unstructured pruning. This figure shows neural network connections before (left) and after pruning (right). Line thickness reflects weight magnitude; dashed lines indicate where low-magnitude weights have been pruned, resulting in a sparser network.

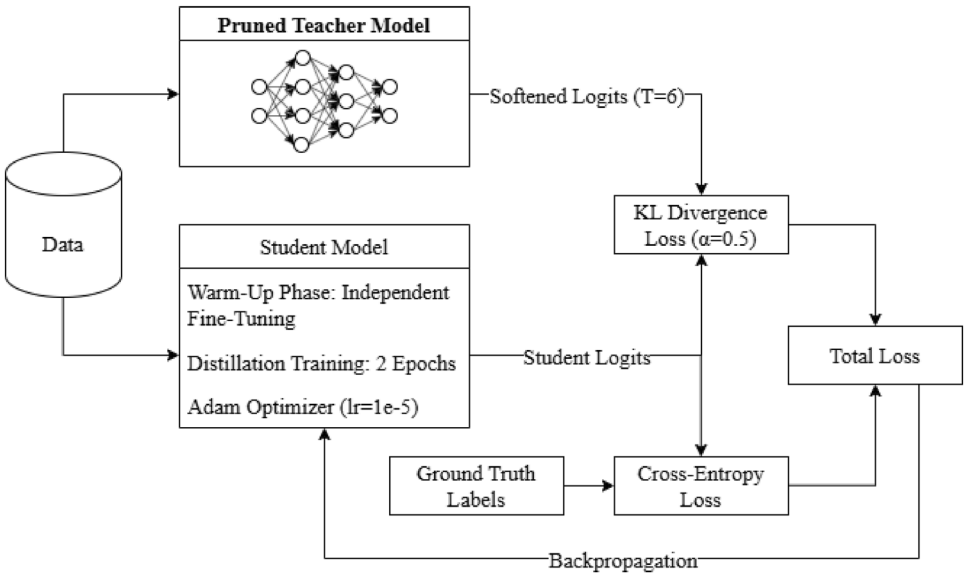

Fig. 11. Knowledge distillation process. This figure shows how the student model is trained by minimizing a combined loss that equally balances KL divergence with the teachers' softened outputs and cross-entropy with ground truth labels.

based on their L1 norm. This pruning rate was chosen as a starting point, and no other rates were tested due to already observing a decrease in accuracy. The changes were then made permanent using the `prune.remove` method. The pruning was applied as follows: By pruning 10% of the weights in the linear layers, we introduced sparsity while maintaining classification performance. However, since a drop in accuracy was already observed at this pruning level, no additional pruning rates were explored to prevent further degradation in performance.

2. Knowledge distillation: Knowledge distillation is a model compression technique that transfers knowledge from a larger teacher model to a smaller student model, allowing the student to approximate the teacher's predictions while being more computationally efficient. Instead of solely relying on one-hot labels, the student model learns from the teacher's softened probabilities (logits), capturing richer information about prediction uncertainty and class relationships. During distillation, the student model is trained using a combination of two loss functions: Kullback–Leibler (KL) divergence loss, which aligns the student's output distribution with the teacher's, and cross-entropy loss, which ensures accurate predictions on ground truth labels. The process often involves a temperature parameter that smooths the logits, improving knowledge transfer efficiency. Recent studies have leveraged knowledge distillation to optimize transformer models, particularly in reducing the computational demands of large language models while retaining their capabilities[6]. In this experiment, knowledge distillation trained a compact DistilBERT student model using a pruned BERT teacher and a pruned ELECTRA teacher. Figure 11 illustrates this process. The teacher models underwent L1 Unstructured Pruning, removing 10% of the lowest-magnitude weights in the linear layers of

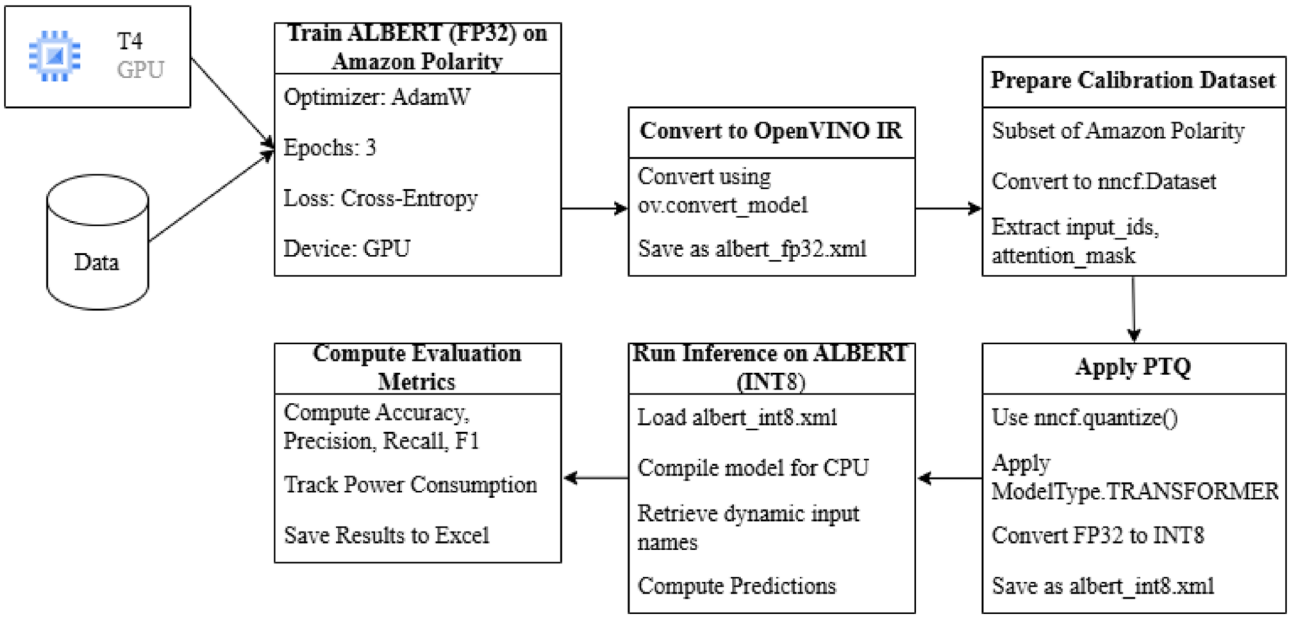

Fig. 12. Post-training quantization process. This figure outlines the process of applying PTQ to the ALBERT model, showing data calibration, OpenVINO IR conversion, and quantization into INT8 format for efficient inference.

the transformer models. Before distillation, the student model was first fine-tuned on the Amazon Polarity dataset for one epoch to establish a baseline understanding. The pruned teachers then generated softened logits, scaled by a temperature parameter $\tau = 6$. This temperature was chosen after testing multiple values ($\tau = 4, 6, 10$) on the BERT model, where lower values led to overfitting and higher values caused overgeneralization, reducing performance. A balanced loss function combined KL divergence and cross-entropy with a weighting factor of $\alpha = 0.5$, ensuring that the student retained essential knowledge from the teacher while still learning from ground truth labels.

3. Quantization: Quantization is a model compression technique that reduces computational demands by representing weights and activations with lower bit precision, such as moving from 32-bit floating point to 8-bit integers[7]. Post-Training Quantization (PTQ) is a method that applies quantization to a pre-trained model without requiring additional training, making it efficient for deployment on hardware with limited computational resources[30]. In our implementation, we used OpenVINO's Neural Network Compression Framework (NNCF) to apply PTQ to an ALBERT model fine-tuned on the Amazon Polarity dataset. First, the full-precision (FP32) model was converted into OpenVINO's Intermediate Representation (IR) format. Then, a small subset of the training data was used as a calibration dataset to determine optimal quantization parameters. The model was then transformed into an INT8 representation using NNCF's quantization API, which replaces high-precision floating-point operations with integer computations, reducing memory footprint and inference latency. The quantized model was compiled and deployed for inference, significantly lowering energy consumption and computational complexity. Figure 12 illustrates the PTQ process applied to our ALBERT model, highlighting conversion, calibration, and quantized inference stages.

Carbon measurement tools

To quantify our AI models' energy consumption and environmental impact, we utilized CodeCarbon, an open-source tool engineered to measure carbon emissions associated with machine learning tasks by tracking hardware power usage and factoring in the carbon intensity of local energy sources[9]. CodeCarbon installation was completed within the Python environment where the models were trained. Through real-time monitoring, a CodeCarbonTracker instance logged energy usage across CPU, GPU, and RAM, enabling an accurate comparison of emissions between baseline and compressed models. The tool automatically detects hardware specifications, and we ensured that it accurately reflected our setup's regional energy grid by verifying the location-based data obtained via the Electricity Map API. If regional carbon intensity data was unavailable, CodeCarbon defaulted to a global average of 475 gCO2/kWh, ensuring a standardized approach when necessary. The emissions rate is calculated using total emissions divided by the total duration of model training and inference:

$$\text{Emissions rate} = \frac{\text{Total } CO_2 \text{ Emissions (kg)}}{\text{Time (s)}} \tag{7}$$

This provides an estimate of how much carbon is emitted per unit of training time. However, in this study CodeCarbon's reporting captures only training and inference time, it does not account for model initialization or dataset loading. To control for hardware variability, all models were trained under the same computational

Metric	Description	Calculation method
Total energy (kWh)	Total energy consumed by CPU, GPU, and RAM	Total energy = CPU energy + GPU energy + RAM energy
Total carbon emissions (kg)	Emissions as carbon-equivalents [carboneq]	Emissions = $C \times E$, where C is the carbon intensity and E is the energy consumed.
CPU energy (kWh)	Energy used by the CPU	Tracked by Intel Power Gadget (Windows/Mac Intel), powermetrics (Apple Silicon), or Intel RAPL files (Linux). Defaults to TDP-based approximation if unavailable. This experiment was done using Linux, so CPU Energy was tracked using Intel RAPL files.
GPU Energy (kWh)	Energy used by GPU	Tracked using Nvidia GPUs via the `pynvml` library
RAM Energy (kWh)	Energy used by RAM	Estimated with a fixed ratio of 3 W per 8 GB of RAM
Emissions Rate (kg/s)	Emissions divided by duration	Emissions Rate = Total Emissions/Duration

Table 2. Metrics and calculation methods used by CodeCarbon.

environment, using an NVIDIA Tesla T4 GPU and Intel Xeon Gold 5218 CPU. Energy consumption across different hardware components was measured at 15-s intervals, preventing discrepancies from short-term fluctuations. GPU power was tracked using pynvml, CPU power using Intel RAPL files, and RAM energy was estimated using CodeCarbon's standard 3W per 8GB ratio. By maintaining a consistent hardware setup and standardized energy tracking methods, we ensured that the differences in energy consumption reflected variations in model efficiency rather than hardware discrepancies. The following table summarizes each metric we measured, the method used to derive it, and the source of the data utilized by CodeCarbon.

Table 2 discusses the metrics utilized for evaluation. These metrics together provide a detailed overview of our models' environmental impact, allowing for comparisons of model efficiency and guiding energy-efficient development practices.

Experimental setup

Training and testing configurations

Training parameters

The model was fine-tuned using the following hyperparameters:

- Batch size: 16
- Learning rate: 2×10^{-5}
- Number of epochs: 3

These hyperparameters were selected based on an initial set of experiments conducted with the BERT model. Specifically, different learning rates (1×10^{-5}, 2×10^{-5}, and 5×10^{-5}) and epoch settings (1, 3, 5, and 10) were tested. The results indicated that 2×10^{-5} provided the best balance between convergence speed and performance, while increasing the learning rate led to instability and a higher risk of overfitting. Similarly, while increasing the number of epochs beyond 3 resulted in marginal accuracy gains, it also significantly increased energy consumption without substantial improvement in performance. The batch size of 16 was chosen to optimize memory efficiency while maintaining stable gradient updates. A larger batch size (e.g., 32) led to higher GPU memory usage, while a smaller batch size (e.g., 8) caused training instability due to noisier gradient estimates. These choices ensure that the models were fine-tuned efficiently while maintaining high accuracy and minimizing unnecessary computational costs.

Hardware configuration

The experiments were carried out on a virtual machine configured with the following specifications:

- Operating System: Linux 6.1.0-25-amd64-x86_64 with glibc 2.36
- CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz (12 physical cores)
- GPU: NVIDIA Tesla T4 with 16 GB of GDDR6 memory
- NVIDIA Driver Version: 560.35.03
- CUDA Version: 12.6
- RAM: 15.64 GB

Model-compression technique combinations

To establish a baseline for our metrics, we fine-tuned a BERT model on a subset of the Amazon Polarity Dataset. Afterward, we employed a combination of model compression techniques across four models: BERT, DistilBERT, ALBERT, and ELECTRA. Figure 13 outlines the workflow for implementing these combinations. L1 Unstructured Pruning and Knowledge Distillation were applied to the BERT model. L1 Unstructured Pruning was applied to the DistilBERT model. Post-Training Quantization was applied to the ALBERT model. L1 Unstructured Pruning and Knowledge Distillation were applied to the ELECTRA model.

Results

The performance and environmental impact of each model was measured using key metrics. In particular, precision, accuracy, recall, F1-score, ROC AUC, training time (minutes), total energy consumption (kWh), total

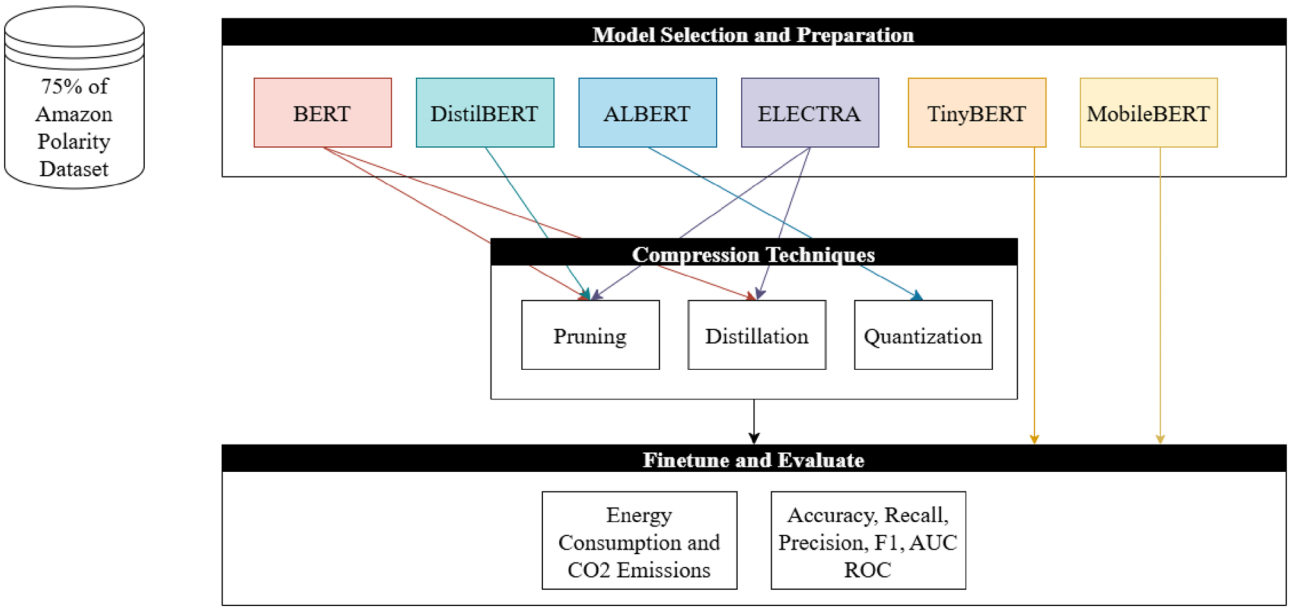

Fig. 13. Workflow of model compression techniques.

carbon emissions (kg carbon), GPU energy, CPU energy, RAM energy, and emissions rate were all measured. Tables 3 and 4 summarize the results, highlighting the trade-offs when using model compression.

To gather information on the classification performance of each model, we calculated their confusion matrices, as shown in Figures 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23. These matrices showcase the distribution of true positive, true negative, false positive, and false negative predictions in each model.

Performance and efficiency analysis

The trade-off between energy consumption and model performance was measured by comparing the accuracy and ROC AUC metrics against the total energy consumed by each model as seen in Fig. 24.

The distribution of energy consumption across each model's CPU, GPU, and RAM components was measured. Figure 25 displays how these distributions differ.

The total amount of carbon emissions for each model was calculated. These values were then compared with each other as seen in Fig. 26.

The relationship between model runtime and total energy consumption was analyzed using a scatter plot. Figure 27 illustrates these trade-offs.

The accuracy and ROC AUC metrics for each model configuration. Figure 28 highlights the consistency of these metrics across different models.

Discussion

This study has explored the impact of L1 Unstructured Pruning, Knowledge Distillation, and Post-Training Quantization on transformer-based models' performance, energy consumption, and carbon emissions. The results show that compression techniques can significantly reduce energy consumption and carbon emissions while maintaining comparable performance for most models. The baseline BERT model exhibited the highest energy consumption and carbon emissions, reflecting the computational intensity of transformer-based models. In contrast, all compressed models tested in this study showed substantial reductions in energy consumption, with energy savings ranging from 2.62 to 91.26% relative to BERT. The total energy consumption of the compressed models varied from 0.628 kWh (TinyBERT) to 7.0525 kWh (ALBERT w/ Quantization). Figure 24 shows that as energy consumption increases, the ROC AUC of the compressed models remains consistently high, ranging between 0.9776 and 0.9930, while accuracy decreases only slightly, with a maximum drop of 0.0025. This suggests that a large increase in energy usage does not yield a proportional improvement in accuracy. Figure 28 further supports this trend, showing that this pattern holds across different model architectures and applied optimizations. Figure 25 highlights the significant reduction in overall energy consumption for compressed models compared to the BERT baseline. Additionally, it shows that the distribution of energy consumption across computing resources remains consistent regardless of the model architecture or applied compression technique. Across all models, the GPU remains the largest contributor to total energy consumption, followed by the CPU and RAM. This finding indicates that while total energy consumption is reduced, the relative proportion of resources used remains unchanged. Figure 26 illustrates the carbon emission reductions achieved through model compression, with DistilBERT w/ Pruning achieving the highest reduction of 1.704 kg CO_2(45.69%) compared to BERT. Similarly, ELECTRA w/ Pruning & Distillation showed a 23.86% decrease in emissions compared to its baseline counterpart, reinforcing the environmental benefits of compression techniques. Figure 27 highlights the trade-off between total runtime and energy consumption, demonstrating a strong

Model	BERT Baseline	BERT w/ Pruning & Distillation	DistilBERT Baseline	DistilBERT w/ Pruning	ALBERT Baseline	ALBERT w/ Quantization	ELECTRA Baseline	ELECTRA w/ Pruning & Distillation	TinyBERT	MobileBERT
Total Energy (kWh)	7.197	4.887	3.3637	3.5893	7.5932	7.0525	6.6075	5.0260	0.6287	3.6625
Total CO2 Emissions (kg)	3.366	2.27	1.5627	1.6673	3.5278	3.2765	3.0695	2.3351	0.2923	1.7015
CPU Energy	3.05	2.1377	1.3753	1.5757	3.5317	3.3055	2.7775	2.1325	0.2790	1.5545
GPU Energy	3.8455	2.5383	1.8497	1.8610	3.7299	3.4365	3.5550	2.6830	0.3233	1.9545
RAM Energy	0.3015	0.2077	0.1390	0.1530	0.3315	0.3105	0.2755	0.2105	0.0270	0.1535
Emissions Rate	1.595×10^{-5}	1.641×10^{-5}	1.545×10^{-5}	1.637×10^{-5}	1.734×10^{-5}	1.720×10^{-5}	1.600×10^{-5}	1.593×10^{-5}	1.628×10^{-5}	1.591×10^{-5}
Training Time (min)	3501.165	2310.686	1691.023	1702.099	3390.906	3174.370	3215.0815	2447.571	298.4453	1788.0555

Table 3. Energy metric averages for baseline and compressed models.

Model	BERT Baseline	BERT w/ Pruning & Distillation	DistilBERT Baseline	DistilBERT w/ Pruning	ALBERT Baseline	ALBERT w/ Quantization	ELECTRA Baseline	ELECTRA w/ Pruning & Distillation	TinyBERT	MobileBERT
Accuracy	0.96154	0.95900	0.95882	0.95872	0.86356	0.65445	0.96699	0.95917	0.95154	0.92461
Precision	0.96155	0.95903	0.95882	0.95872	0.89430	0.67820	0.96699	0.95918	0.95154	0.92463
Recall	0.96154	0.95900	0.95882	0.95872	0.86356	0.65445	0.96699	0.95917	0.95154	0.92461
F1	0.96154	0.95900	0.95881	0.95871	0.83751	0.63458	0.96699	0.95917	0.95154	0.92461
ROC AUC	0.99137	0.98871	0.99063	0.99062	0.93708	0.72305	0.99304	0.99041	0.98876	0.97759

Table 4. Performance metric averages for models.

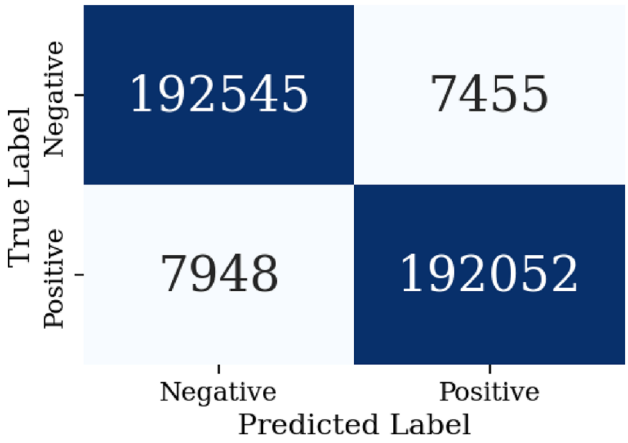

Fig. 14. Confusion matrix for BERT Baseline.

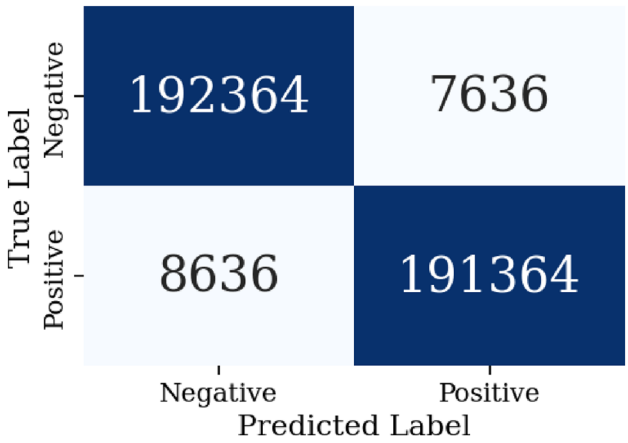

Fig. 15. Confusion matrix for BERT with Pruning and Distillation.

linear correlation with a fit line equation of $y = 0.0022x - 0.1454$ and a correlation coefficient of 0.9952. This indicates that longer training durations are directly proportional to higher energy consumption, reinforcing the importance of optimizing both time and energy usage in AI models.

Unexpected findings in pruning and quantization

Despite the overall trend of significant energy savings with compressed models, we observed an unexpected anomaly in the case of DistilBERT w/ Pruning, where energy consumption increased from 3.363 to 3.589 kWh instead of decreasing. This counterintuitive result may be explained by the inherent overhead associated with non-structured pruning, where the irregular sparsity pattern forces hardware to manage additional index lookups and irregular memory accesses. Such overheads can diminish the computational savings that pruning is expected to offer. These findings align with prior research, which suggests that while non-structured pruning enables a higher pruning rate, it often incurs storage and computational inefficiencies due to extra index handling

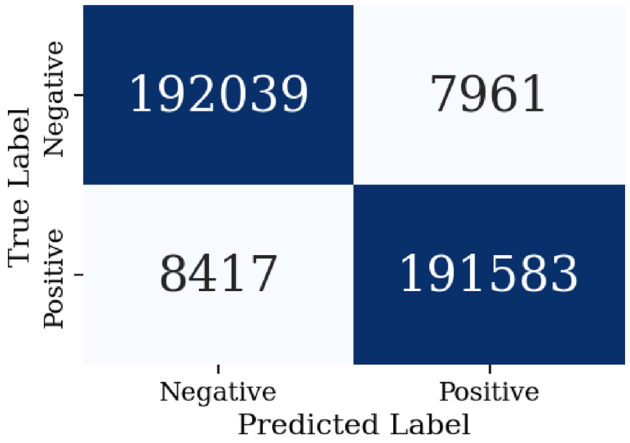

Fig. 16. Confusion matrix for DistilBERT Baseline.

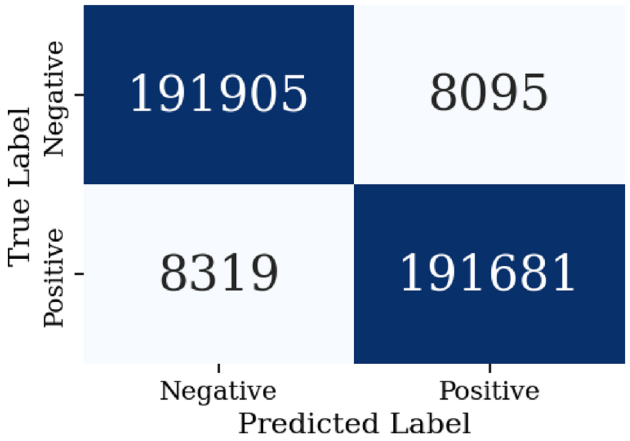

Fig. 17. Confusion matrix for DistilBERT with Pruning.

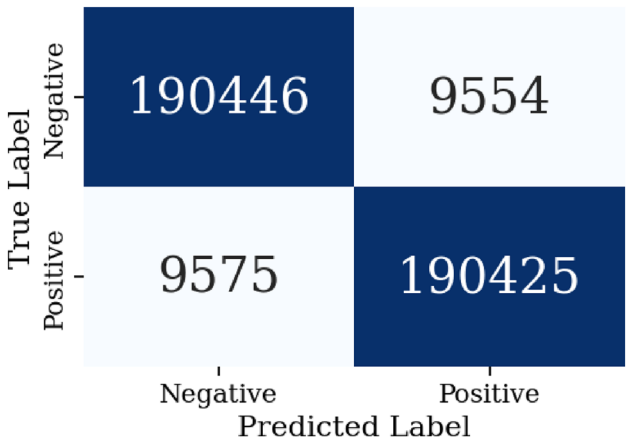

Fig. 18. Confusion matrix for ALBERT Baseline.

required for sparse matrix operations[35]. This suggests that structured pruning methods may be a more effective alternative for reducing energy consumption without introducing additional computational overhead.

Similarly, ALBERT w/ Quantization exhibited a drastic performance drop, with accuracy decreasing to 65.44% and an F1-score of 63.46%, despite reducing energy usage by 7.12% compared to ALBERT Baseline. The decline in performance can be attributed to quantization-induced precision loss in ALBERT's already

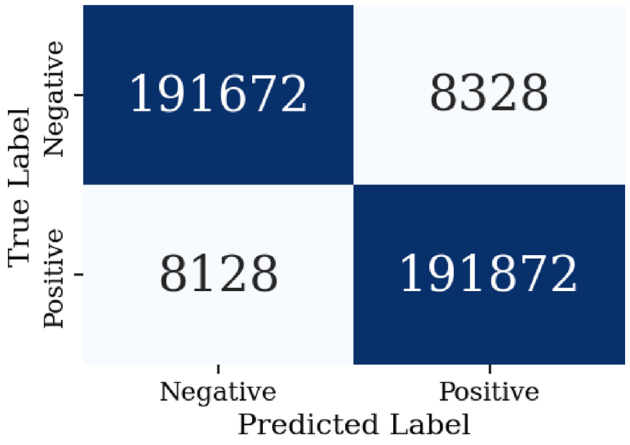

Fig. 19. Confusion matrix for ALBERT with Quantization.

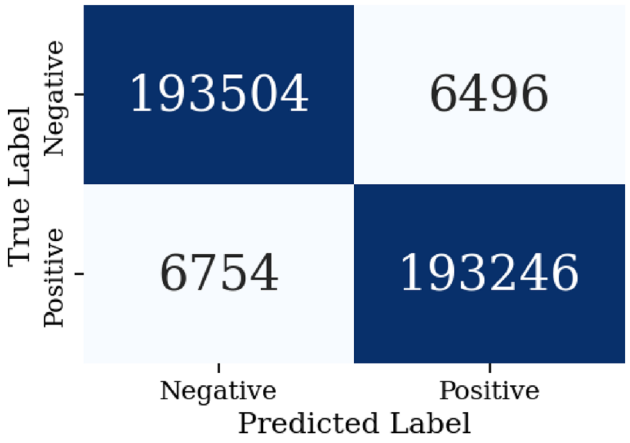

Fig. 20. Confusion matrix for ELECTRA Baseline.

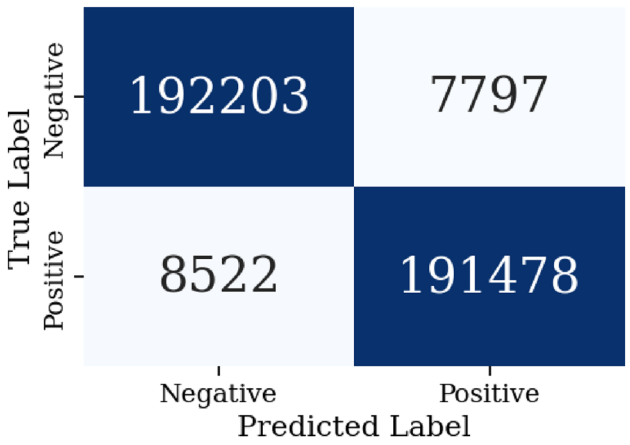

Fig. 21. Confusion matrix for ELECTRA with Pruning and Distillation.

highly compressed architecture. According to prior research, post-training quantization (PTQ) disrupts the self-attention mechanism by altering the ranking of attention weights, leading to misaligned feature prioritization[36]. Additionally, high inter-channel variance in LayerNorm makes it highly sensitive to quantization errors, which reduces the model's expressiveness and precision. Unlike quantization-aware training (QAT), PTQ lacks the ability to retrain and adjust for these distortions, further exacerbating performance degradation. These findings

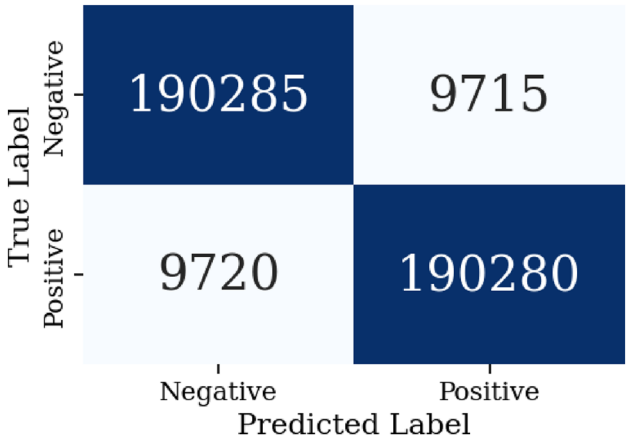

Fig. 22. Confusion matrix for TinyBERT.

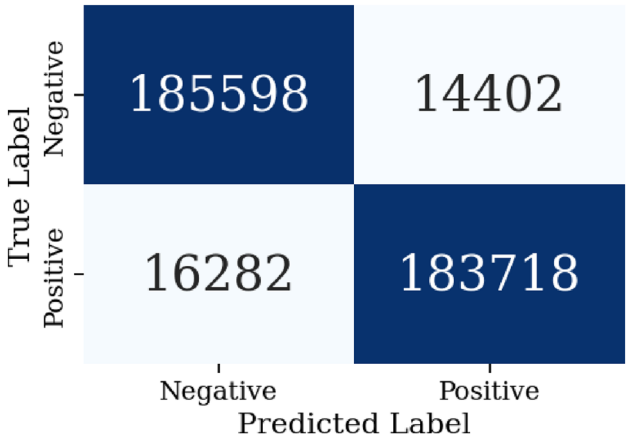

Fig. 23. Confusion matrix for MobileBERT.

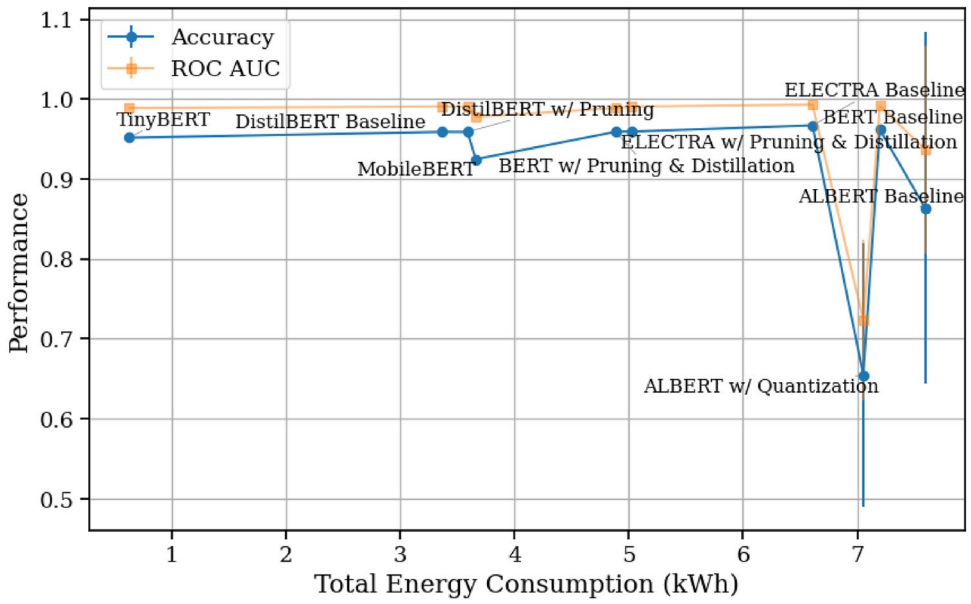

Fig. 24. Scatter plot of energy consumption vs. performance trade-offs. Shows that increases in energy usage do not yield proportionally higher accuracy or ROC AUC, indicating diminishing returns in performance.

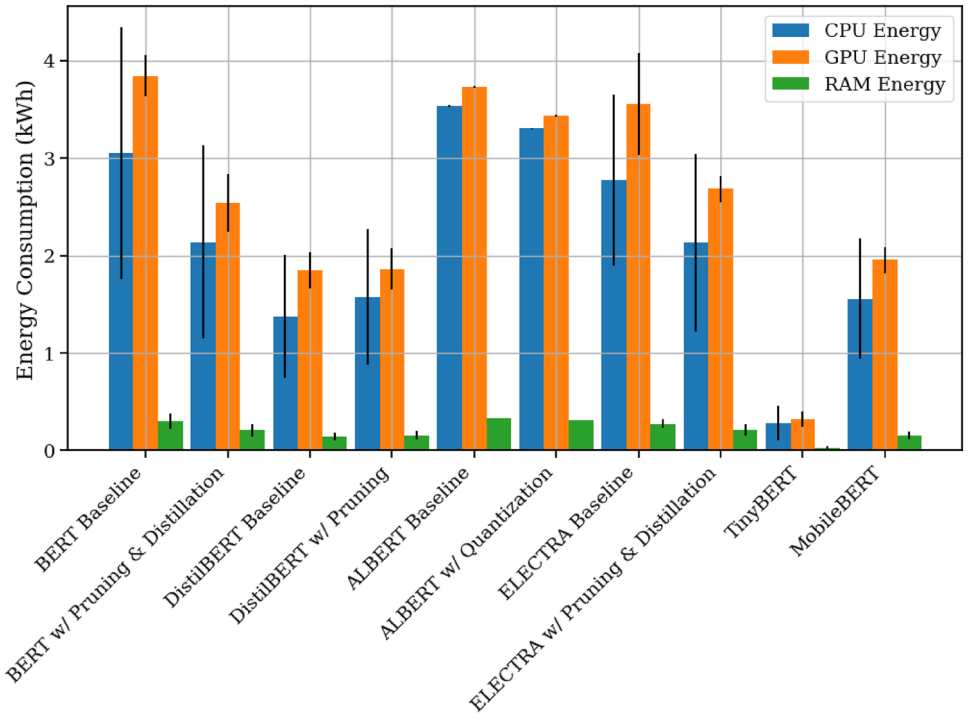

Fig. 25. Grouped bar chart of total energy consumption of CPU, GPU, and RAM components. Shows that the GPU is the primary contributor to total energy usage across all models, followed by CPU and RAM.

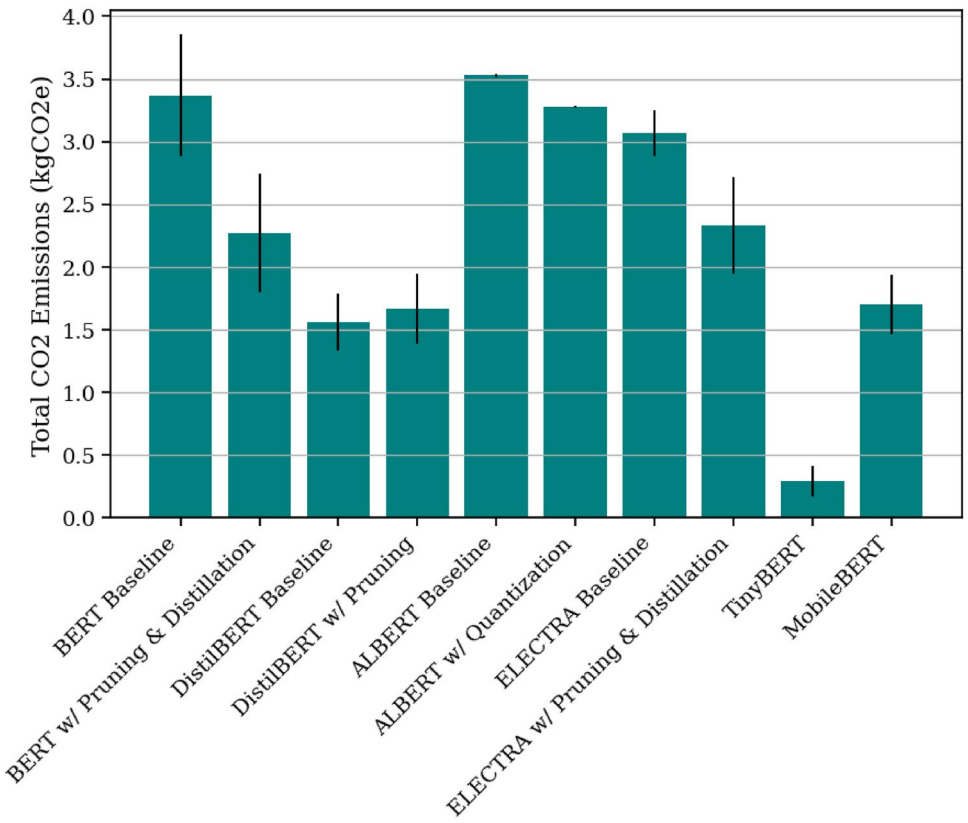

Fig. 26. Bar chart of carbon emissions by model. Shows that compressed models, particularly BERT w/ Pruning & Distillation and ELECTRA w/ Pruning & Distillation, achieve notable emission reductions.

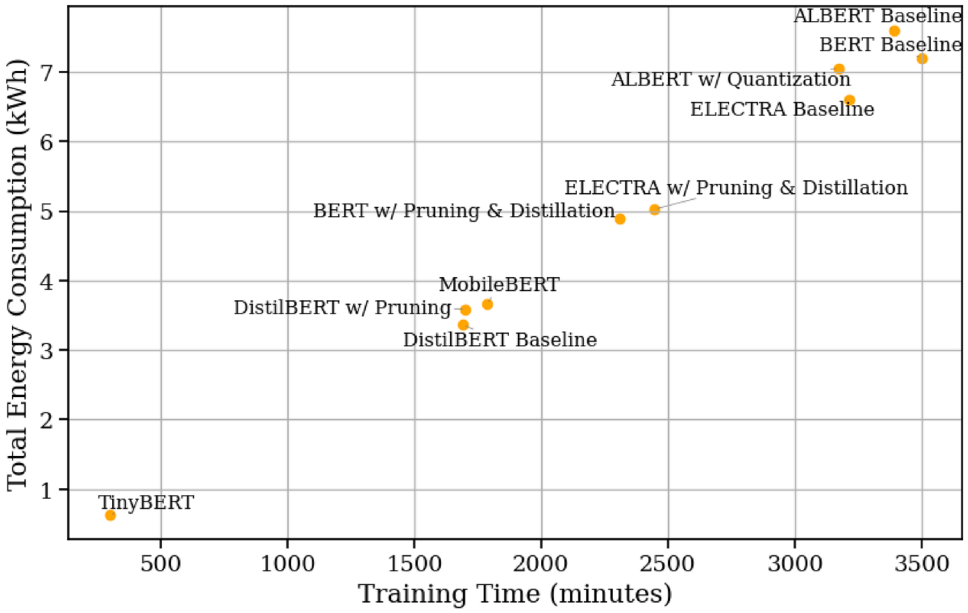

Fig. 27. Scatter plot of runtime vs. energy consumption. Shows a strong linear correlation between runtime and energy usage, emphasizing the importance of time-efficient configurations.

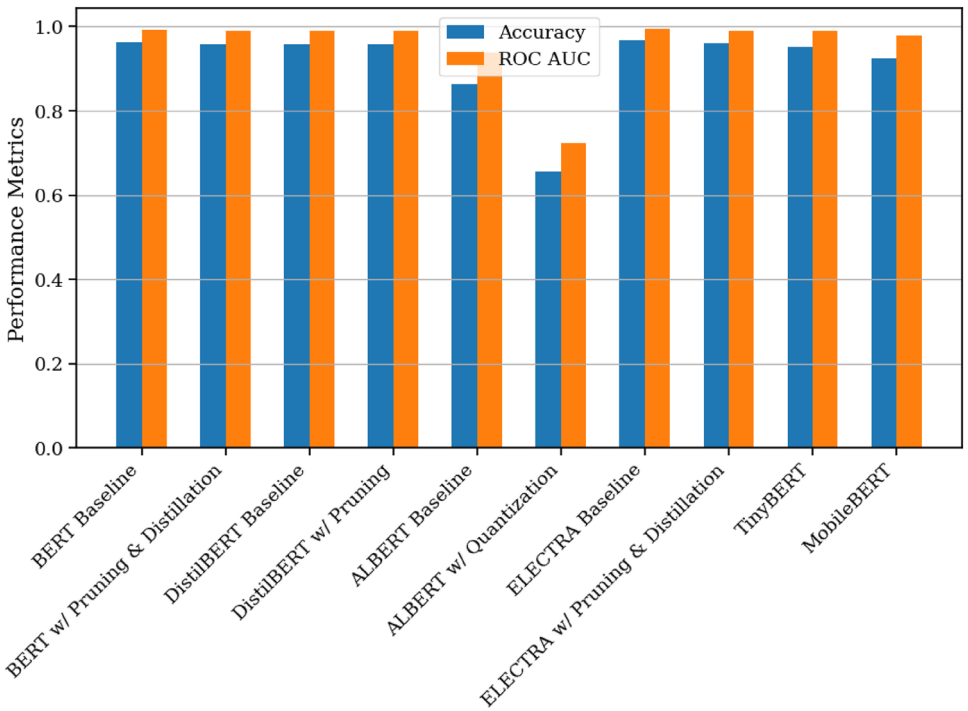

Fig. 28. Grouped bar chart of performance metrics across models. Shows consistently high accuracy and ROC AUC across most of the models, supporting the reliability of compressed configurations.

suggest that alternative quantization strategies, such as mixed-precision quantization or alternative scaling methods, may be necessary to mitigate accuracy loss in PTQ applications.

Statistical analysis

To further evaluate the variability in energy consumption and performance metrics, we conducted a statistical analysis based on standard deviation and 95% confidence intervals (CI), presented in Table 5. The results indicate that TinyBERT exhibited the lowest variability in energy consumption (SD = 0.1032 kWh), whereas ALBERT models showed minimal variance (SD = 0.0092 kWh and 0.0085 kWh for baseline and quantized versions,

Model	Metric	Standard Deviation	95% CI Lower	95% CI Upper
BERT Baseline	Total Energy (kWh)	0.467	6.04	8.36
	Total CO2 Emissions (kg)	0.194	2.88	3.85
	CPU Energy	0.518	1.76	4.34
	GPU Energy	0.0845	3.64	4.06
	RAM Energy	0.0335	0.22	0.38
	Training Time (min)	75.422	3313.81	3688.52
BERT w/ Pruning & Distillation	Total Energy (kWh)	0.4047	3.88	5.89
	Total CO2 Emissions (kg)	0.1881	1.80	2.74
	CPU Energy	0.3976	1.15	3.13
	GPU Energy	0.1194	2.24	2.83
	RAM Energy	0.0268	0.14	0.27
	Training Time (min)	122.096	2007.38	2613.99
DistilBERT Baseline	Total Energy (kWh)	0.1943	2.88	3.85
	Total CO2 Emissions (kg)	0.0904	1.34	1.79
	CPU Energy	0.2543	0.74	2.01
	GPU Energy	0.0751	1.66	2.04
	RAM Energy	0.0156	0.10	0.18
	Training Time (min)	76.840	1500.14	1881.91
DistilBERT w/ Pruning	Total Energy (kWh)	0.2435	2.98	4.19
	Total CO2 Emissions (kg)	0.1129	1.39	1.95
	CPU Energy	0.2802	0.88	2.27
	GPU Energy	0.0832	1.65	2.07
	RAM Energy	0.0178	0.11	0.20
	Training Time (min)	75.144	1515.43	1888.77
ALBERT Baseline	Total Energy (kWh)	0.0092	7.57	7.62
	Total CO2 Emissions (kg)	0.0038	3.52	3.54
	CPU Energy	0.0037	3.52	3.54
	GPU Energy	0.0049	3.72	3.74
	RAM Energy	0.0005	0.33	0.33
	Training Time (min)	3.651	3381.84	3399.98
ALBERT w/ Quantization	Total Energy (kWh)	0.0085	7.03	7.07
	Total CO2 Emissions (kg)	0.0035	3.27	3.29
	CPU Energy	0.0035	3.30	3.31
	GPU Energy	0.0045	3.43	3.45
	RAM Energy	0.0005	0.31	0.31
	Training Time (min)	3.418	3165.88	3182.86
ELECTRA Baseline	Total Energy (kWh)	0.1595	6.21	7.00
	Total CO2 Emissions (kg)	0.0745	2.88	3.25
	CPU Energy	0.3525	1.90	3.65
	GPU Energy	0.2110	3.03	4.08
	RAM Energy	0.0185	0.23	0.32
	Training Time (min)	210.475	2692.23	3737.93
ELECTRA w/ Pruning & Distillation	Total Energy (kWh)	0.3340	4.20	5.86
	Total CO2 Emissions (kg)	0.1551	1.95	2.72
	CPU Energy	0.3655	1.22	3.04
	GPU Energy	0.0550	2.55	2.82
	RAM Energy	0.0235	0.15	0.27
	Training Time (min)	49.279	2325.16	2569.99
TinyBERT	Total Energy (kWh)	0.1032	0.37	0.88
	Total CO2 Emissions (kg)	0.0478	0.17	0.41
	CPU Energy	0.0719	0.10	0.46
	GPU Energy	0.0320	0.24	0.40
	RAM Energy	0.0050	0.01	0.04
	Training Time (min)	29.930	224.09	372.80

Continued

Model	Metric	Standard Deviation	95% CI Lower	95% CI Upper
MobileBERT	Total Energy (kWh)	0.2095	3.14	4.18
	Total CO2 Emissions (kg)	0.0975	1.46	1.94
	CPU Energy	0.2475	0.94	2.17
	GPU Energy	0.0535	1.82	2.09
	RAM Energy	0.0155	0.11	0.19
	Training Time (min)	57.784	1644.51	1931.60

Table 5. Energy metric variability for baseline and compressed models.

Model	BERT Baseline vs. BERT w/ Pruning & Distillation	DistilBERT Baseline vs. DistilBERT w/ Pruning	ALBERT Baseline vs. ALBERT w/ Quantization	ELECTRA Baseline vs. ELECTRA w/ Pruning & Distillation
Accuracy p-value	0.01406	0.11841	0.14563	0.00009
Precision p-value	0.01233	0.11841	0.06662	0.00009
Recall p-value	0.01406	0.11841	0.14563	0.00009
F1 p-value	0.01372	0.10905	0.22330	0.00009
ROC AUC p-value	0.02595	0.75382	0.05633	0.00039

Table 6. Statistical significance (p-values) of performance metric comparisons.

respectively), suggesting stable but high energy consumption. BERT models displayed wider confidence intervals, with total energy consumption ranging from 6.04 to 8.36 kWh, indicating greater variability.

Statistical significance tests (Table 6) show that:

- BERT w/ Pruning & Distillation significantly differs from BERT Baseline in accuracy ($p = 0.01406$), F1-score ($p = 0.01372$), and ROC AUC ($p = 0.02595$), confirming that pruning and distillation introduce a minor but measurable performance degradation.
- DistilBERT w/ Pruning showed no statistically significant difference from its baseline across all metrics ($p > 0.1$), suggesting that pruning had a negligible effect on performance.
- ELECTRA w/ Pruning & Distillation exhibited the most statistically significant impact, with all p-values < 0.001, indicating a measurable trade-off between accuracy and energy savings.
- ALBERT w/ Quantization did not show statistical significance for accuracy ($p = 0.14563$) but had a notable decline in F1-score ($p = 0.22330$), aligning with the observed performance drop due to quantization errors.

These results emphasize that while compression techniques reduce energy consumption, they may introduce minor but statistically significant performance trade-offs, depending on the model architecture and applied optimization method.

Trade-offs between compression techniques

Each compression technique presents distinct advantages and drawbacks, making their effectiveness highly dependent on model architecture and application constraints.

- Knowledge Distillation (BERT w/ Pruning & Distillation, ELECTRA w/ Pruning & Distillation): These models demonstrated substantial energy savings while maintaining strong performance. ELECTRA w/ Pruning & Distillation reduced energy consumption by 23.86% compared to its baseline, making it a viable alternative for efficiency-focused applications.
- L1 Unstructured Pruning (DistilBERT w/ Pruning): While pruning typically aims to reduce computational costs, DistilBERT w/ Pruning consumed slightly more energy (3.589 kWh) than its baseline (3.364 kWh), likely due to irregular sparsity overhead, suggesting that structured pruning may be more effective.
- Post-Training Quantization (ALBERT w/ Quantization): ALBERT w/ Quantization exhibited the most significant performance drop, with an F1-score decline of 23.46%, despite reducing energy usage by 7.12% compared to ALBERT Baseline. This suggests that quantization may not be suitable for already compressed architectures.
- Baseline Transformer Models (BERT, DistilBERT, ELECTRA, ALBERT): These models offered the highest accuracy but at the cost of high energy consumption, reinforcing their computational intensity.
- Carbon-Efficient Transformers (TinyBERT, MobileBERT): These models achieved the lowest energy consumption and carbon emissions, with TinyBERT using just 0.629 kWh (91.26% less than BERT Baseline), making them ideal for energy-constrained environments.

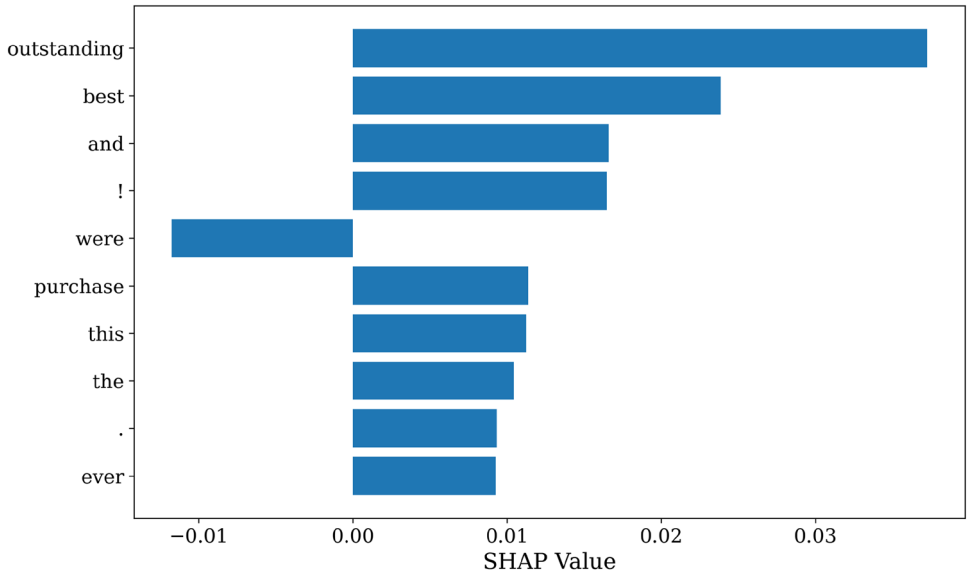

Fig. 29. SHAP Bar Plot for TinyBERT. Highlights modest but focused token contributions, supporting the model's interpretability alongside lowest energy use.

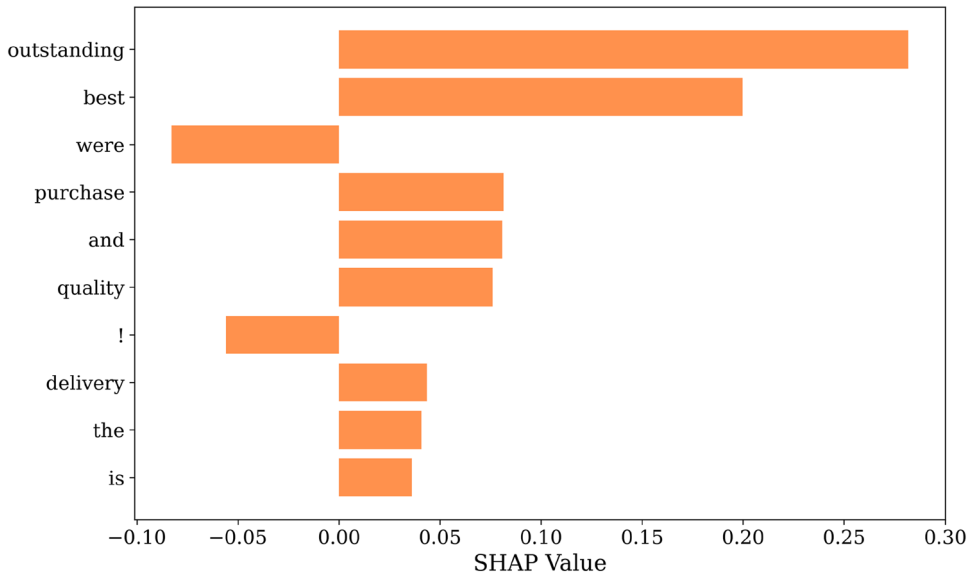

Fig. 30. SHAP Bar Plot for ELECTRA with Pruning & Distillation. Shows strong emphasis on sentiment-heavy tokens like "outstanding" and "best", aligning with the model's high accuracy.

Overall, distillation proved to be the most balanced technique, offering strong energy savings with minimal performance trade-offs, while pruning and quantization introduced energy and accuracy trade-offs that depend on model architecture.

SHAP analysis

To improve the interpretability of the most energy-efficient and highest-performing compressed models, we conducted a SHAP (SHapley Additive exPlanations) analysis to evaluate token-level contributions to sentiment classification. We selected TinyBERT, which showed the lowest total energy consumption (0.629 kWh), and ELECTRA with Pruning and Distillation, which achieved the highest accuracy (0.959) among all compressed models. Using a representative input sentence, "This is the best purchase I've ever made. Quality and delivery were outstanding!", we computed SHAP values for each token. Figure 29 display bar plots of the top 10 most influential tokens from TinyBERT whereas Fig. 30 represents bar plots of the top 10 most influential tokens from ELECTRA with Pruning & Distillation model. Both models predicted a positive sentiment, with TinyBERT returning a confidence score of 0.9968 and ELECTRA with Pruning and Distillation returning 0.7422. In TinyBERT, the

Study	Accuracy	Precision	Recall	F1-score
AlQahtani et al.[31]	0.947	0.946	0.947	0.946
Mustafaa and Elsaid[32]	0.940	0.940	0.930	0.930
Alkarawi and Al-Hafidh[33]	0.935	0.9352	0.9351	0.9350
Hashmi et al.[34]	0.890	0.880	0.890	0.880
Our Study: BERT Baseline	0.962	0.962	0.962	0.962
Our Study: BERT with pruning and distillation	0.959	0.959	0.959	0.959

Table 7. Performance comparison of BERT-based sentiment analysis models on Amazon reviews.

most influential tokens included "best" (0.0239), "outstanding" (0.0373), and "and" (0.0166), while tokens like "were" had a negative influence (-0.0118), indicating the learned importance of positive sentiment cues with minimal energy expenditure. In contrast, ELECTRA with Pruning and Distillation exhibited more sharply defined importance scores, with "outstanding" contributing a value of 0.2818, "best" at 0.1999, and "quality" at 0.0761, reinforcing its strong classification performance.

These results correspond with the quantitative metrics and offer model-specific insights into token-level decision-making, supporting the observation that compressed models not only maintain predictive performance but also preserve important feature recognition.

Broader implications of model compression

In standard sentiment analysis applications, such as customer feedback classification, product review aggregation, or social media monitoring, a drop in of 0.27 in accuracy or a 0.18 drop in ROC AUC is unlikely to cause significant misclassification issues, given that the overall accuracy remains above 92.3% across all models (Table 4). Additionally, Table 7 compares our BERT-based sentiment analysis models with prior research, demonstrating their superior accuracy and F1-score. In these domains, small trade-offs in predictive performance may be acceptable, especially when energy efficiency is a priority, such as in large-scale deployments with constrained computational resources. However, for high-precision NLP tasks where even marginal performance drops can have critical consequences, full-scale models may still be preferable despite their higher energy demands.

AI compression techniques are crucial for large-scale AI deployments, balancing computational efficiency and accuracy. Companies such as Amazon, McDonald's, and Netflix use AI-powered sentiment analysis to process large amounts of user-generated content, with Amazon using it for product reviews and McDonald's for customer feedback monitoring[37]. In such cases, slight accuracy trade-offs are acceptable for significant energy savings. Beyond sentiment analysis, BrainBox AI optimizes HVAC systems in commercial buildings, reducing energy consumption by up to 25% and carbon emissions by 40%[38]. These findings align with this study, which demonstrates how compressed AI models support energy efficiency without compromising performance. In finance, BlackRock and Siemens use AI for real-time risk assessment and forecasting, with Siemens improving financial reporting accuracy by 10% using AI-driven models[39]. Given the real-time nature of financial modeling, reducing computational costs while maintaining integrity further validates the benefits of model compression. These examples highlight the widespread impact of energy-efficient AI, from e-commerce and entertainment to finance and sustainability.

In addition to the energy and performance metrics reported in this study, our analysis reveals that compression techniques can also influence the internal behavior of transformer-based models, particularly in their attention mechanisms and feature importance. Recent surveys on transformer compression[40] and on model compression for large language models[41] have demonstrated that compression-especially non-structured pruning-can alter the distribution of attention weights, potentially diminishing the interpretability of certain attention heads. These works indicate that while many attention heads may be redundant and can be pruned with minimal performance loss, the irregular sparsity introduced by non-structured methods may disrupt the regular patterns of feature importance that are critical for maintaining interpretability. Our findings suggest that even when overall accuracy remains stable, compression might shift which features the model relies on, underscoring the need for deeper analyses of attention dynamics in compressed models.

Strengths and limitations

This study provides insights into the impact of model compression techniques on energy-efficient AI. One of its key strengths is the comprehensive evaluation of pruning, knowledge distillation, and quantization in multiple models, allowing a detailed comparison of their effectiveness in reducing energy consumption and carbon emissions. Additionally, including TinyBERT and MobileBERT, which are naturally compact models without applied compression techniques, provides a valuable reference point for understanding the trade-offs between pre-designed lightweight and compressed models. The Amazon Polarity Dataset ensures real-world applicability, as sentiment analysis is a widely used task in NLP with significant commercial and research relevance. By integrating CodeCarbon to monitor energy consumption and emissions, this study provides empirical evidence of the trade-offs between model performance and sustainability, contributing to the growing field of Green AI. The variety of model architectures explored, including BERT, DistilBERT, ALBERT, ELECTRA, TinyBERT, and MobileBERT, further enhances the applicability of our findings. Finally, the results demonstrate that significant reductions in energy consumption can be achieved with minimal impact on accuracy, reinforcing the viability of model compression techniques for sustainable AI development.

Despite these strengths, this study has several limitations. The primary limitation is its dataset scope, as the findings are based on the Amazon Polarity Dataset, which, while suitable for sentiment analysis, may not generalize to other NLP tasks such as machine translation, question answering, or biomedical text processing. Another limitation is that the study does not separately analyze energy consumption during training and inference; distinguishing between these phases would provide a more granular understanding of energy efficiency in real-world deployments. Additionally, all experiments were conducted on an NVIDIA Tesla T4 GPU, which may not reflect energy efficiency trends on more advanced hardware such as NVIDIA A100 GPUs or Google's TPUs. Furthermore, while three widely used model compression techniques were explored, other techniques such as structured pruning, low-rank factorization, and hybrid quantization could offer further efficiency improvements. Although TinyBERT and MobileBERT were included for comparison, these models were designed through proprietary distillation techniques, making their exact compression methodologies less transparent than explicitly applying pruning, quantization, or knowledge distillation in this study. The reported energy readings using CodeCarbon do not account for model loading time, training begins. This can lead to slight underestimations of total energy consumption. Additionally, CodeCarbon relies on hardware-dependent power estimation techniques, which can introduce inaccuracies. For example, energy readings for CPUs and GPUs depend on vendor-provided APIs and may not fully reflect dynamic power scaling mechanisms. Energy consumption estimates may vary significantly in cloud-based or multi-GPU environments due to workload scheduling and virtualization overhead. While this study focuses on sentiment analysis, the impact of compression techniques on high-precision tasks, such as financial sentiment analysis or medical text classification, remains an open question. In such domains, even small reductions in accuracy or recall could introduce higher risks, requiring further evaluation before deployment. Finally, the study does not incorporate a cost analysis, meaning the trade-off between energy savings and potential computational overhead introduced by compression techniques remains unexamined. Addressing these limitations in future research could lead to a more comprehensive understanding of sustainable AI practices and further optimize model compression strategies.

Future work

This study demonstrates the potential of model compression techniques to reduce energy consumption and carbon emissions in transformer models. However, several directions remain for future research. One key area is separating energy consumption during training and inference. A more granular analysis could help optimize efficiency for real-world deployment. Expanding the scope of datasets and tasks is another crucial step. While this study focuses on sentiment analysis, applying compression techniques to other NLP tasks, such as machine translation or question answering, would provide a broader perspective. Extending this work to speech and vision models could also uncover new insights into compression effectiveness across modalities. A notable limitation of the current study is that it does not explore the impact of compression techniques on the internal mechanisms of transformer models, such as feature importance and attention distributions. Future research should incorporate analyses using attention visualization tools and quantitative feature importance metrics to assess how different compression strategies affect these internal representations. Such investigations could provide additional insights into the trade-offs between energy efficiency, model interpretability, and performance. Further exploration of compression strategies is needed. Beyond L1 Unstructured Pruning, Knowledge Distillation, and Post-Training Quantization, techniques such as structured pruning, low-rank factorization, and hybrid quantization could enhance efficiency. Investigating dynamic or adaptive compression methods could provide models that effectively balance performance and energy efficiency. Hardware considerations also play a crucial role. This study was conducted using an NVIDIA Tesla T4 GPU, but testing on more advanced architectures like A100 GPUs, TPUs, or dedicated AI accelerators could reveal additional energy savings and help identify hardware-specific optimizations. Finally, incorporating a cost-benefit analysis would provide a more practical perspective. While this study focuses on energy reduction, understanding the trade-offs between computational savings, model accuracy, and financial costs would be valuable for real-world adoption. Future research in these areas can further bridge the gap between AI performance and sustainability, ensuring models remain effective and environmentally responsible.

Conclusion

The findings from the experiments reinforce the observation that there is potential in using model compression techniques, such as L1 Unstructured Pruning, Knowledge Distillation, and Post-Training Quantization, to make transformer-based models more environmentally sustainable. By applying these techniques to BERT, DistilBERT, ALBERT, and ELECTRA, we achieved notable reductions in energy usage, with the most significant energy savings observed in TinyBERT, which reduced energy consumption by 91.26% compared to BERT. Additionally, by incorporating TinyBERT and MobileBERT as inherently efficient baselines, we gained insights into whether explicit compression methods outperform models specifically designed for energy efficiency. Our findings show that accuracy remained within 92.461–95.917% across all compressed models, demonstrating that compression can achieve substantial energy savings with minimal accuracy degradation. BERT with Pruning and Distillation and ELECTRA with Pruning and Distillation exhibited the most significant reductions in total energy consumption (2.31 kWh and 1.582 kWh, respectively) and carbon emissions (1.096 kg CO_2 and 0.734 kg CO_2, respectively), suggesting that these combinations are particularly effective for sustainable AI practices. Furthermore, we observed that GPU energy remains the dominant contributor to total energy usage across all models, reinforcing the need for compression techniques that specifically target GPU-intensive computations. From a practical standpoint, the results suggest that compressed models provide a cost-effective and environmentally conscious alternative to traditional full-scale transformer models for applications in sentiment

analysis, customer feedback classification, and large-scale NLP tasks. The relatively small trade-offs in accuracy (≤ 0.008) and ROC AUC (≤ 0.003) in models where distillation was applied indicate that compressed models remain viable for real-world deployment, particularly in energy-constrained settings. This study contributes to the ongoing efforts to align machine learning advancements with sustainability goals by demonstrating that compression techniques can meaningfully reduce AI's environmental footprint. These findings highlight the importance of integrating Green AI practices into future AI model development, ensuring that deep learning remains powerful and environmentally responsible.

Data availability

The datasets generated and/or analysed during the current study are available at the following GitHub repository: https://github.com/eileenpaula/Achieving-Carbon-Efficient-AI.

Received: 9 January 2025; Accepted: 17 June 2025
Published online: 02 July 2025

References

1. Dhar, P. The carbon impact of artificial intelligence. *Nat. Machine Intell.* **2**, 423–425. https://doi.org/10.1038/s42256-020-0219-9 (2020).
2. Rafat, K. et al. Mitigating carbon footprint for knowledge distillation based deep learning model compression. *PLOS One* **18**, e0285668. https://doi.org/10.1371/journal.pone.0285668 (2023).
3. Malihi, L. & Heidemann, G. Efficient and controllable model compression through sequential knowledge distillation and pruning. *Big Data Cognitive Comput.* **7**, 154. https://doi.org/10.3390/bdcc7030154 (2023).
4. Argerich, M. F. & Patiño-Martínez, M. Measuring and improving the energy efficiency of large language models inference. *IEEE Access* **12**, 67890–67904. https://doi.org/10.1109/ACCESS.2024.3409745 (2024).
5. Cheng, H., Zhang, M. & Shi, J. Q. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations (2024). arxiv:2308.06767.
6. Bergmann, D. What is knowledge distillation? (2024).
7. Nagel, M. *et al.* A white paper on neural network quantization (2021). arxiv:2106.08295.
8. Wang, H. & Fu, Y. Epsd: Early pruning with self-distillation for efficient model compression. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (2023).
9. CodeCarbon. Codecarbon: Track and reduce carbon emissions from machine learning computing (2020). Available at: https://github.com/mlco2/codecarbon.
10. Anthony, L. F. W., Kanding, B. & Selvan, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint* arXiv:2007.03051 (2020).
11. Henderson, P. et al. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Machine Learn. Res.* **21**, 1–43 (2020).
12. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (2020). arxiv:1910.01108.
13. Lan, Z. *et al.* Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations* (2020).
14. Clark, K., Luong, M.-T., Le, Q. V. & Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations* (2020).
15. Khan, S. et al. Optimizing deep neural network architectures for renewable energy forecasting. *Discov. Sustain.* **5**, 394 (2024).
16. Khan, S., Mazhar, T., Khan, M. A. & et al. Comparative analysis of deep neural network architectures for renewable energy forecasting: Enhancing accuracy with meteorological and time-based features. *Discov. Sustain.* **5**, 533. https://doi.org/10.1007/s43621-024-00783-5 (2024).
17. Inam, S. A., Khan, A. A., Mazhar, T. & et al. Pr-fcnn: a data-driven hybrid approach for predicting pm2.5 concentration. *Discov. Artif. Intell.* **4**, 75, https://doi.org/10.1007/s44163-024-00184-7 (2024).
18. Inam, S. A. et al. A novel deep learning approach for investigating liquid fuel injection in combustion system. *Discov. Artif. Intell.* **5**, 32. https://doi.org/10.1007/s44163-025-00248-2 (2025).
19. Khan, M. A. et al. Optimizing smart home energy management for sustainability using machine learning techniques. *Discov. Sustain.* **5**, 430 (2024).
20. Tabbakh, A. et al. Towards sustainable AI: A comprehensive framework for green AI. *Discov. Sustain.* https://doi.org/10.1007/s43621-024-00641-4 (2024).
21. Liu, V. & Yin, Y. Green AI: Exploring carbon footprints, mitigation strategies, and trade offs in large language model training. *Discov. Artif. Intell.* https://doi.org/10.1007/s44163-024-00149-w (2024).
22. Jiao, X. *et al.* Tinybert: Distilling bert for natural language understanding (2020). arxiv:1909.10351.
23. Sun, Z. *et al.* Mobilebert: a compact task-agnostic bert for resource-limited devices (2020). arxiv:2004.02984.
24. Liang, T., Glossner, J., Wang, L., Shi, S. & Zhang, X. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing.* https://doi.org/10.1016/j.neucom.2021.07.045 (2021).
25. Guo, M. et al. Pruning and quantization algorithm with applications in memristor-based convolutional neural network. *Cognit. Neurodyn.* **18**, 233–245 (2024).
26. Fred, L. *Amazon Review Polarity* https://doi.org/10.6084/m9.figshare.13232501.v1 (2020).
27. Rogers, A., Kovaleva, O. & Rumshisky, A. A primer in bertology: What we know about how bert works (2020). arxiv:2002.12327.
28. Vaswani, A. *et al.* Attention is all you need (2023). arxiv:1706.03762.
29. Kim, T., Oh, J., Kim, N., Cho, S. & Yun, S.-Y. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation (2021). arxiv:2105.08919.
30. Li, J., Zhang, T., Yen, I. E.-H. & Xu, D. Fp8-bert: Post-training quantization for transformer (2023). arxiv:2312.05725.
31. AlQahtani, A. S. M. Product sentiment analysis for amazon reviews. *International Journal of Computer Science and Information Technology* (2021).
32. Mostafa, M. & AlSaeed, A. Sentiment analysis based on bert for amazon reviewer. *J. ACS Adv. Comput. Sci.* **13**, 1–10. https://doi.org/10.21608/asc.2023.171559.1007 (2022).
33. Al-Hafidh, N. & Alkarawi, A. Advanced sentiment analysis of amazon electronics reviews leveraging bert: Model optimization and evaluation. *Procedia Comput. Sci.* **258C**, 3608–3618. https://doi.org/10.1016/j.procs.2025.04.616 (2025).
34. Ali, H., Hashmi, E., Yildirim Yayilgan, S. & Shaikh, S. Analyzing amazon products sentiment: A comparative study of machine and deep learning, and transformer-based techniques. *Electronics* **13**, 1305. https://doi.org/10.3390/electronics13071305 (2024).
35. Ma, X. *et al.* Non-structured dnn weight pruning–is it beneficial in any platform? *IEEE Trans. Neural Netw. Learn. Syst.* **PP**, 1–15, https://doi.org/10.1109/TNNLS.2021.3063265 (2021).

36. Liu, Z., Wang, Y., Han, K., Ma, S. & Gao, W. Post-training quantization for vision transformer (2021). arxiv:2106.14156.
37. Ciufo, N. 10 real-world examples of ai-powered sentiment analysis (2024).
38. Staff, A. A new virtual ai assistant built on amazon bedrock can reduce the carbon footprint of commercial buildings. here's how. (2024).
39. Solutions, C. Ai in financial modeling and forecasting: 2024 guide: Coherent solutions (2025).
40. Tang, Y. *et al.* A survey on transformer compression (2024). arxiv:2402.05964.
41. Zhu, X., Li, J., Liu, Y., Ma, C. & Wang, W. A survey on model compression for large language models (2024). arxiv:2308.07633.

Acknowledgements

This research is supported by US-Department of Energy-Environmental Management (DOE-EM) (DE-EM0005213).

Author contributions

E.P. and J.S. conceived the experiment, E.P. conducted the experiment, and E.P. and J.S. analyzed the results. All authors reviewed the manuscript.

Additional information

Correspondence and requests for materials should be addressed to J.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.