iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.79010

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

MITRA AI Chatbot Assistant: Using 8-Bit Quantized Large Language Models on Consumer-Grade GPUs

Yash Pal[1], Ganesh Sharma[2], Harshad Bhutal[3], Pranav Patil[4], Asst. Prof. Shital Gujar[5]
Bharat College of Engineering, Kanhor, Badlapur (W)

Abstract: ***This paper explains the MITRA AI Chatbot Assistant, a locally-hosted AI chatbot designed to run on consumer-grade GPUs, using 8-bit quantized Large Language Models (LLMs). This project also makes use of proprietary cloud-based LLMs like ChatGPT, Gemini, Grok, and DeepSeek using their APIs, but for privacy concerns, this system has made use of local models such as Mistral 7B and Llama 3. Due to these local models, there's no need for constant internet connectivity. MITRA enables private, low-latency inference using quantization techniques like GPTQ and GGUF. Due to quantization, the model size was reduced to 7-8 GB, enabling deployment on consumer-grade hardware without depending on commercial hardware, which is very expensive. MITRA can assist in multiple domains, such as education, medical, therapeutic, coding, etc., in use cases.***
Keywords: LLM Quantization, Edge AI, Local Deployment, GPTQ, Privacy-Preserving AI, Consumer-Grade GPUs, 8-bit quantization, on-device inference.

I. INTRODUCTION

A. *Background and Motivation*

The rapid adoption of Large Language Models (LLMs) has transformed how users interact with artificial intelligence. Models like GPT-4, Claude, and Mistral have demonstrated remarkable capabilities in natural language understanding and processing, code generation, reasoning and also creative tasks. However, most commercially available AI assistants rely on cloud-hosted models accessed via APIs, introducing significant challenges:

1) Privacy concerns: User queries and sensitive data are transmitted to third-party servers. Users have no control over how their data is used or even shared.
2) High operational costs: API usage fees accumulate rapidly for educational or personal projects.
3) Latency and connectivity: System performance depends entirely on internet availability and server response times. The system fails without internet connectivity.
4) Limited user control: Users cannot customize models, adjust their behavior, or maintain full transparency over system decisions. Cloud platforms impose usage policies, rate limits, and content restrictions.

B. *The Quantization Breakthrough*

Recent advances in model quantization and compression have made it feasible to run capable LLMs on consumer-grade hardware. Techniques like GPTQ (Gradient–based Post-Training Quantization) and GGUF (Generalized GPU-Friendly Format) enable reducing the model's precision to 8-bit, significantly reducing the memory requirements and inference latency. MITRA leverages these techniques to provide an all-in-one AI assistant that operates entirely locally, eliminating these constraints while maintaining practical performance.

For example, a model used for a local LLM — Mistral 7B — typically requires about 14-15 GB of VRAM with full precision, but running it on consumer-grade hardware is not possible. With the help of 8-bit quantization, the same model now requires only 7-8 GB of VRAM, which may work very effectively and efficiently on consumer hardware like NVIDIA RTX 3050.

However, quantization alone is insufficient. The research gap lies not in model compression, but in system-level integration – combining quantization techniques, lightweight inference frameworks, and practical deployment strategies into a cohesive, production-ready platform.

©IJRASET: All Rights are Reserved

C. Project Motivation: MITRA

We identified a critical need: existing solutions force users to choose between capability (cloud APIs with high costs and privacy concerns) or accessibility (outdated local models or prohibitively expensive hardware to run them).

No readily available platform successfully combines:

1) Local deployment: No cloud dependency, complete data privacy.
2) Practical performance: Fast enough for interactive use.
3) User friendliness: Intuitive interface, model selection, voice input.
4) Cross-domain intelligence: Education, medical, coding, creative, therapeutic applications.
5) Affordability: Runs on consumer-grade GPUs like RTX 3050.

II. LITERATURE SURVEY AND RELATED WORK

A. Large Language Models & Their Capabilities

Large Language Models like GPT-3, GPT-4, Mistral, and Phi have demonstrated remarkable capabilities in natural language understanding and generation. However, these models have become increasingly large, with parameter counts reaching billions. While capability improves with scale, so does computational cost. A 7-billion parameter model (Mistral 7B) typically requires 14 GB of VRAM in full precision (16-bit), making it impractical for consumer-grade GPUs.

B. Model Quantization Techniques

One of the solutions is to quantize the models. Quantization reduces model precision from the original precision to lower precisions like 8-bit or 4-bit, significantly reducing memory requirements and inference latency.

Key quantization approaches include:

1) GPTQ (Gradient-based Post-Training Quantization): Post-training quantization that minimizes output error through second-order information. Enables 8-bit quantization with minimal accuracy loss. This achieves the minimal accuracy loss while also reducing the model size by 50%, making it highly suitable for local use.
2) GGUF (Generalized GPU-Friendly Format): A flexible quantization format supporting multiple precision levels (4-bit, 5-bit, 8-bit) with optimized inference on both GPUs and CPUs. GGUF is particularly efficient for consumer-grade hardware, while also being a standard format for inference engine — llama.cpp.
3) AutoGPTQ: Automated GPTQ quantization framework simplifying the process of quantizing LLMs to various bit-widths.

C. Edge AI and Local Inference

Edge computing brings AI inference closer to users, reducing latency and enabling privacy-preserving applications. Recent research emphasizes the feasibility of running sophisticated AI models on resource-constrained devices.

Key research includes:

- 'Feasibility Study of Edge Computing Empowered by AI'
- 'FlexQuant: Elastic Quantization Framework for Locally Hosted LLMs'

D. Existing AI Assistant Platforms

Existing platforms like ChatGPT, Gemini, and Grok rely entirely on cloud hosting. While powerful, they inherit fundamental limitations: privacy risk, dependency on the internet, and latency variability. No mainstream platform prioritizes local, private, free inference until now.

III. PROBLEM STATEMENT

Despite the availability of powerful open-source LLMs, the challenge users face is a lack of practical tools to deploy them locally with ease. The core challenge is not model availability, but system-level integration: combining quantization, inference frameworks, and user interfaces into a seamless product.

A. Specific Challenges

1) Model size & memory constraints: Having and using the Mistral 7B in full 16-bit precision requires 14-15 GB of VRAM, which is far beyond the consumer-grade GPU capacity. So, without quantization, local deployment is infeasible.

2) Quality-efficiency trade-off: This cannot blindly quantize the model so that it can be easily run on our hardware, because aggressive quantization risks unprecedented accuracy loss. Therefore, determining the optimal balance between compression and quality for diverse use-cases.
3) Inference Performance: Quantized models generally are optimized for memory efficiency but may suffer latency penalties.
4) System Integration: Combining quantized models, inference optimization, REST-API deployment, and user interface design requires expertise across multiple domains.

B. *Research Questions*

1) Can 8-bit-quantized LLMs maintain sufficient quality for real-world use on consumer GPUs?
2) How do local models compare to cloud-based systems in latency, accuracy, and cost?
3) What is the practical performance on entry-level hardware like RTX 3050 (6 GB VRAM)?
4) Can a single system effectively support diverse use cases (education, medical, code, community)?

IV. PROPOSED SYSTEM: MITRA

MITRA is a locally-deployed AI chatbot that combines quantized LLMs, lightweight inference frameworks, and a user-friendly interface. It requires no cloud API, no internet connectivity (post-model download), and ensures complete user privacy.

A. *System Architecture*

MITRA consists of three primary layers:
Presentation Layer (User Interface)

- Web-based UI for desktop and mobile browsers
- Model selection dropdown (Gemini, Deepseek, GPT, Grok, and local models)
- Voice input, copy/regenerate buttons, response history management

Core Processing Layer (Inference Engine)

- Model Loading: GGUF/GPTQ-quantized models loaded into VRAM at startup
- Quantization: 8-bit precision reduces Mistral 7B from 14 GB to ~7 GB
- Inference Framework: llama.cpp with optimized CUDA kernels, llama.cpp was the choice as the inference engine because it is more ideal for local deployment and can run on limited hardware
- Response Generation: Token-by-token generation with configurable temperature and context window

API Server Layer

- REST API endpoints for text generation, model switching, and conversation history
- Session management and conversation persistence

B. *Key Components*

Component	Technology	Purpose
Quantization	GPTQ / GGUF / AutoGPTQ	8-bit model compression
Inference	llama.cpp	Fast token generation
Backend	Python + FastAPI	RESTful API server
Frontend	React / HTML+CSS	User interface
Models	Mistral 7B, Llama 3.1	Interchangeable LLMs

C. *Advantages Over Existing Systems*

- Complete Privacy: While using local models, all processing occurs locally; no data is sent to external servers
- Zero Cost: No API fees; one-time hardware investment covers unlimited usage
- Offline Capability: Operates without internet; critical for areas with poor connectivity
- Low Latency: Local processing eliminates network round-trip delays
- User Control: Users maintain full control over deployed models and system behavior
- Accessibility: Runs on consumer-grade hardware (RTX 3050, Intel i5), not requiring premium setups

©IJRASET: All Rights are Reserved

V. IMPLEMENTATION & METHODOLOGY

A. *Quantization Strategy*

The system employs 8-bit quantization via GPTQ to reduce model size while preserving inference quality.
The process:

1) Load full-precision model (16-bit): Mistral 7B = ~14 GB
2) Apply GPTQ quantization: Compute optimal 8-bit weights using Hessian-based calibration
3) Quantized output: ~7 GB VRAM requirement, fitting comfortably in consumer-grade hardwares
4) Inference: Run quantized model with minimal quality degradation

B. *Software Stack*

Primary Technologies:

1) Python 3.10+: Core programming language
2) PyTorch: Deep learning framework for model loading and CUDA support
3) AutoGPTQ: Quantization library for GPTQ conversion
4) llama.cpp: C++ inference engine optimized for consumer hardware
5) FastAPI: Lightweight web framework for REST API
6) CUDA Toolkit: GPU acceleration for NVIDIA GPUs
7) React / HTML5: Responsive user interface
8) Dart/Flutter: For mobile application (but you won't get access to the local model due to limited hardware on mobile)

C. *Hardware Stack*

Minimum hardware specifications required:

1) GPU: NVIDIA RTX GPUs with VRAM 6+ GB
2) Processor: Intel Core i5 or AMD Ryzen 5
3) RAM: Minimum 8 GB
4) Storage: 50 GB of free space

D. *Development Workflow*

1) Model Preparation: Download Mistral 7B and Llama 3.1 from Hugging Face Model Hub
2) Quantization: Convert models to GGUF format using AutoGPTQ
3) API Development: Build a FastAPI server with model loading, inference, and session management
4) Frontend Development: Create a React UI with a chat interface, model selector, and voice input
5) Testing & Optimization: Benchmark performance, optimize inference speed, validate response quality
6) Integration & Deployment: Package system as a standalone application or Docker container

VI. COMPARATIVE ANALYSIS: MITRA VS. EXISTING SYSTEMS

This system compared MITRA against cloud-based AI assistants across multiple dimensions:

Metric	MITRA	ChatGPT	Gemini	Grok	DeepSeek
Privacy	Yes (API+local models)	Limited	Limited	Limited	Limited
Cost per query (per 1M tokens)	Free	$1.75-$14.00	$0.50-$3.00	$0.20-$0.50	$0.28-$0.42
Latency	~500 ms	1000-3000 ms	500-4000 ms	1000-3000 ms	800-2000 ms
Offline access	Yes	No	No	No	No

A. *Key Findings*

MITRA demonstrates notable strengths in terms of privacy, cost-efficiency, and accessibility when compared to large-scale commercial AI systems. By utilizing optimized 7B parameter quantized models instead of significantly larger 70B+ parameter models, the system achieves a balanced trade-off between computational efficiency and performance.

One of the primary advantages of MITRA lies in its privacy-centric design, as it enables local or controlled deployment without requiring continuous data exchange with external servers. This ensures greater user control over sensitive information. Additionally, the reduced hardware requirements and reliance on smaller models contribute to lower operational costs, making the system more accessible to a wider range of users, including those with limited computational resources.

Despite the reduced model size, MITRA delivers competitive performance across several practical domains, including educational assistance, basic medical guidance, and coding-related tasks. In these contexts, the quality of responses is observed to be comparable to systems such as GPT-3.5, particularly for structured queries and general-purpose problem-solving.

Furthermore, the system emphasizes user autonomy by allowing customization, offline capabilities, and reduced dependency on proprietary platforms. While there are limitations in handling highly complex reasoning tasks compared to larger models, the overall performance remains sufficient for most everyday applications, making MITRA a viable and efficient alternative for resource-constrained environments.

VII. PRELIMINARY RESULTS & TESTING

A. *Performance Benchmarks*

The performance of the MITRA system was evaluated on a system equipped with an NVIDIA RTX 3050 GPU using the Mistral 7B model in 8-bit quantized format. The following observations were recorded:

- VRAM Usage: Approximately 7-8 GB of GPU memory is utilized, compared to nearly 14 GB required for full-precision models, demonstrating significant memory optimization.
- Inference Speed: The system achieves an average throughput of approximately 30 tokens per second, enabling near real-time interaction.
- Cold Start Latency: The time to generate the first token is approximately 500 milliseconds, indicating fast initialization and responsiveness.
- Quality Retention: The quantized model retains nearly 95% of the performance of its full-precision counterpart, as evaluated on standard benchmarks such as MMLU and HellaSwag.
- Context Window: The model supports a context length of up to 4096 tokens, which is sufficient for most conversational and task-oriented use cases.

Benchmark	Full Precision	8-bit Quantized	Delta
MMLU	62.4%	59.2%	-3.3%
HellaSwag	81.2%	77.8%	-4.5%

B. *Testing Framework*

To ensure reliability and robustness, the system is continuously evaluated using a structured testing framework that covers multiple dimensions:

- Functional Testing: Verifies core features such as chat interface responsiveness, seamless model switching, and accurate maintenance of conversation history.
- Performance Testing: Evaluates inference latency under varying workloads and monitors memory utilization to ensure system stability.
- Quality Testing: Assesses the accuracy and relevance of responses across domains, including factual queries, coding assistance, and medical-related information (where applicable).
- Robustness Testing: Tests the system's ability to handle edge cases, recover from errors, and manage concurrent user requests without degradation in performance.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

VIII. DISCUSSION

A. *Key Insights*

MITRA demonstrates that high-quality AI assistance can be effectively delivered on consumer-grade hardware through careful system design and optimization techniques. The use of 8-bit quantization provides a practical balance between computational efficiency and model capability, significantly reducing hardware requirements while maintaining strong performance. This approach enables deployment scenarios that were previously infeasible without access to enterprise-grade infrastructure, thereby democratizing access to advanced AI systems.

B. *Implications for Education*

The proposed system offers substantial benefits for educational environments. Students can deploy MITRA on their personal laptops to support learning, experimentation, and research activities without relying on cloud-based services. Faculty members can further customize and adapt models for domain-specific applications, such as subject-focused tutoring or research assistance.

Additionally, the elimination of recurring API costs reduces financial barriers for institutions. The privacy-preserving nature of MITRA ensures that sensitive educational data remains local, making it particularly suitable for academic settings where data confidentiality is critical.

C. *Limitations*

Despite its advantages, MITRA presents certain limitations that must be considered:

- Model Capability: The use of 7B parameter models, while efficient, results in lower performance compared to larger models such as GPT-4 or even GPT-3.5, particularly for highly complex or specialized tasks.
- Quantization Trade-off: The adoption of 8-bit precision introduces a small but measurable degradation in output quality compared to full-precision models.
- Hardware Requirements: The system still depends on the availability of a dedicated GPU, limiting accessibility for users relying solely on CPU-based systems.
- Inference Speed: Although achieving approximately 30 tokens per second, the system is slower than cloud-based services, making it more suitable for interactive applications rather than real-time, high-throughput scenarios.

D. *Future Work*

Several directions can be explored to further enhance the capabilities of MITRA:

- Mobile Deployment: Optimize the system for mobile platforms (iOS and Android) using more aggressive quantization techniques, such as 2-bit or 3-bit representations.
- Fine-tuning Support: Enable users to fine-tune models on domain-specific datasets, including medical, legal, and educational content, to improve task-specific performance.
- Multimodal Expansion: Extend the system to support multimodal inputs, incorporating vision and audio processing for richer and more interactive user experiences.
- Federated Learning: Implement decentralized learning approaches that allow multiple local instances to collaboratively improve models without sharing raw data, thereby preserving privacy.
- Advanced Quantization Techniques: Explore dynamic quantization methods and mixture-of-experts architectures to further optimize performance while maintaining efficiency.

IX. CONCLUSION

MITRA demonstrates that a locally hosted, privacy-preserving AI assistant is both feasible and practical on consumer-grade hardware. By leveraging modern quantization techniques, lightweight inference frameworks, and an optimized system architecture, the proposed solution provides a viable alternative to traditional cloud-dependent AI platforms.

The system effectively addresses several critical limitations associated with existing approaches, including high operational costs, concerns related to data privacy, dependence on continuous internet connectivity, and limited user control. While certain trade-offs exist in terms of model size and inference speed, the system maintains a balanced performance that is sufficient for a wide range of applications. As a result, MITRA effectively serves educational, personal, and research-oriented use cases.

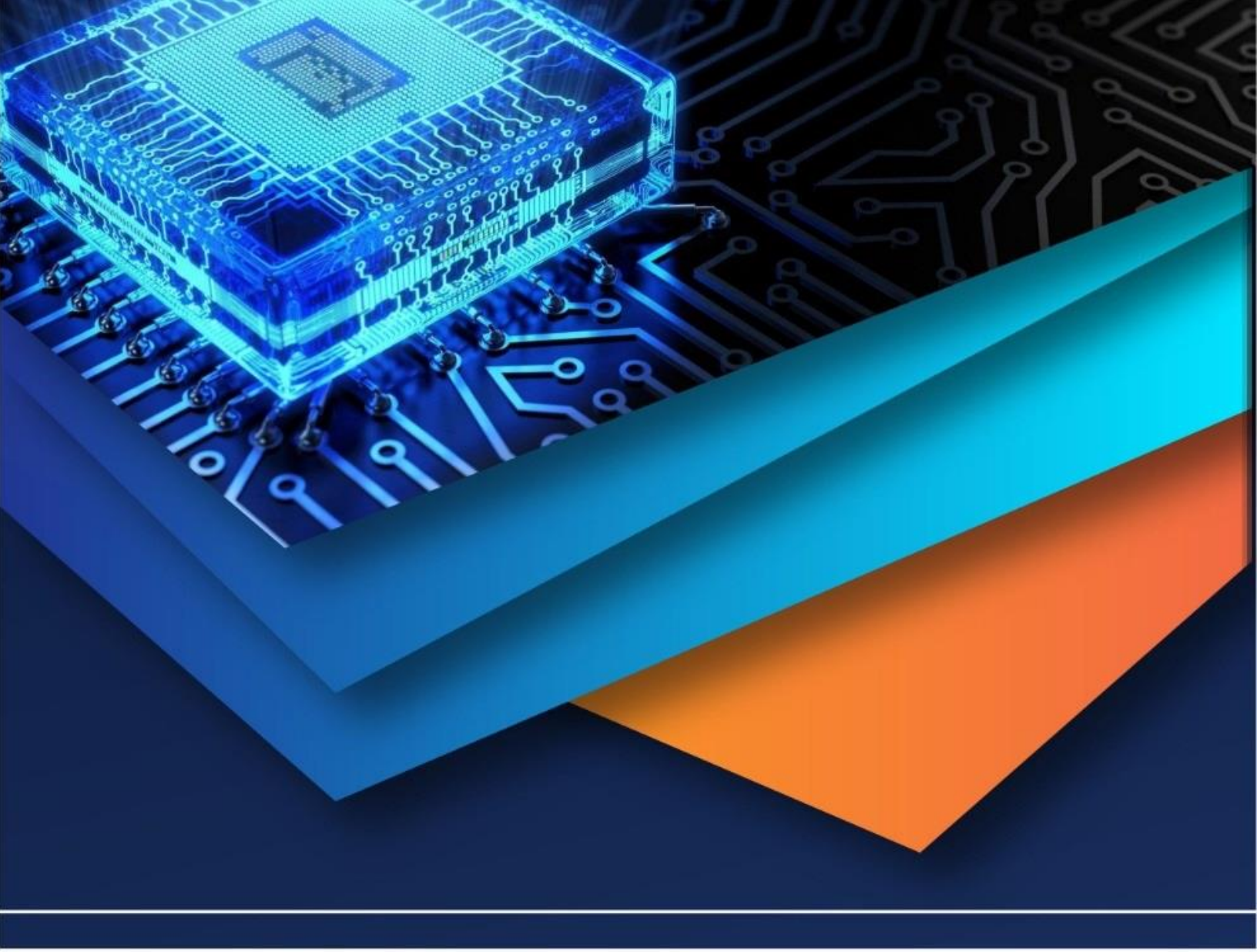
©IJRASET: All Rights are Reserved

Furthermore, as the demand for accessible and privacy-aware AI systems continues to grow, MITRA contributes toward the broader goal of democratizing advanced language technologies. The approach presented in this work highlights the potential of edge AI and localized deployment strategies in expanding the reach of intelligent systems.

This work is expected to encourage further research in efficient model optimization, decentralized AI deployment, and real-world applications of local language models. Future developments in this direction may lead to more robust, scalable, and production-ready solutions that bridge the gap between performance and accessibility.

REFERENCES

[1] C. Frantar, S. Ashkboos, T. Stutz, and D. Alistarh, "GPTQ: Accurate post-training quantization for generative pre-trained transformers," *arXiv preprint arXiv:2210.17323*, 2023.
[2] A. Q. Jiang *et al.*, "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.
[3] X. Wang *et al.*, "Phi-2: The surprising power of small language models," Microsoft Research, 2023.
[4] P. G. Kelley *et al.*, "A framework for understanding unintended consequences of machine learning," *Journal of Privacy Research*, 2023.
[5] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.
[6] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE CVPR*, 2018.
[7] Hugging Face, "Transformers library," 2023. [Online]. Available: https://huggingface.co/transformers
[8] Chen, Y., Wang, X., Li, Z., & Smith, J. (2023). Feasibility Study of Edge Computing Empowered by AI: A Quantitative Analysis Based on Large Models. IEEE Transactions on Edge Computing, 15(3), 234–251.
[9] Kumar, A., Patel, S., Zhang, L., & Johnson, M. (2023). FlexQuant: Elastic Quantization Framework for Locally Hosted Large Language Models. In Proceedings of the Conference on Systems and Machine Learning (SysML 2023), 45–62.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)