LLMScan.online

AI Model Security Scan Report

C

Score 64/100

Report summary

GGUF v3 model | architecture: llama | quantization: F32 | 9 tensors | 2 blocking issues / 2 warnings

Model

Filename	`extreme-dim-tensor.gguf`
Format	`gguf`
Architecture	`llama`
Size	`5.2 KB`
SHA-256	`f5ac9723401ca961624fd671b51fd31c39bbd62c43929e7061f06bad9fc82791`
Scan mode	`static-deep`
Scanned at	`2026-08-10T19:43:42+00:00Z`
Report ID	`05f829a7891c480c91445979c089307c`

Findings (4)

SEVERITY	CATEGORY	DETAIL
HIGH	Tensor Integrity	**Tensors with extreme dimension values (1 tensors)** Examples: [('blk.0.attn_k.weight', 2000000000)] **Recommendation:** Extreme dimension values can cause integer overflow in element-count calculations, enabling heap OOB reads
HIGH	Resource Limits	**Implausible compression ratio: 0.000 bits/param** File size 5,312 B for ~16,000,000,840 parameters implies <0.5 bpp, which is physically impossible for neural weights **Recommendation:** Tensor count or dimensions are likely inflated; this model may be designed to exhaust RAM when loaded
LOW	Supply Chain	**No license field in GGUF metadata (general.license)** Absence of license information makes it impossible to determine usage rights **Recommendation:** Add general.license to model metadata before distributing; use an SPDX identifier
LOW	Supply Chain	**No embedded provenance metadata** None of general.author, general.source.url, general.base_model.0, or general.source.huggingface.repository are embedded in the GGUF file. Without any origin signal the model cannot be traced to a trusted source or compared against a reference checkpoint for tampering detection. For files uploaded directly, provenance is especially critical because the scanner cannot infer repository origin from the file alone. **Recommendation:** Add at least one of: general.author, general.source.huggingface.repository, or general.source.url

Runtime compatibility

RUNTIME	SUPPORTED	NOTES
llama.cpp	Yes	GGUF v3 container: supported | quant F32: supported

Ollama	Yes	GGUF via llama.cpp backend: container and quantization supported
LM Studio	Yes	GGUF container and quantization supported; GPU/CPU inference
vLLM	No	GGUF not supported; requires safetensors format
HuggingFace Transformers	No	GGUF not supported; requires safetensors or PyTorch checkpoint