DeepSeek-R1, CodeLlama, and WizardCoder Robustness on Out-of-Distribution MLOps Tasks

Assignee Research

May 31, 2026

Abstract

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: How robust are the code adaptation capabilities of DeepSeek-R1, CodeLlama, and WizardCoder when evaluated on out-of-distribution MLOps tasks, and how do their performance metrics (e.g., pass@k,. This paper explores the possibilities of the current generation of Large Language Models for incorporating Machine Learning Operations (MLOps) functionalities into ML training code bases. We evaluate the performance of OpenAI (gpt-3.5-turbo) and WizardCoder (open-source, 15B. 12 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.2/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Automating Code Adaptation for MLOps – A Benchmarking Study on LLMs. Research question: How robust are the code adaptation capabilities of DeepSeek-R1, CodeLlama, and WizardCoder when evaluated on out-of-distribution MLOps tasks, and how do their performance metrics (e.g., pass@k, execution accuracy) differ across varying levels of task complexity?.

2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.2/10.

3 Results

13 papers retrieved. 12 claims extracted; 0 independently verified. Quality review score: 4.2/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
The study selects code examples based on popular frameworks including PyTorch, Keras, sklearn, and PyTorch Lightning.	×	0.03
The dataset includes code examples featured in tutorials offered by library developers.	×	0.03
The dataset complexity ranges from simple models like Basic Convnets and Decision Trees to complex models like Transform	×	0.04
The code examples in the dataset range in length from 20 lines to 800 lines.	×	0.04
The Inlining task methodology involves prompt tuning, temperature sampling, and a DocPrompting method.	×	0.03
The Translation task methodology involves a Data Curation Pipeline and a Prompt Construction Pipeline.	×	0.03
For the MLflow component with the OpenAI model, the performance score is 100 across temperatures 0, 0.2, and 1.	×	0.06
For the Keras component with the WizardCoder model, the performance scores are 50, 25, and 75 at temperatures 0, 0.2, an	×	0.05
For the Sklearn component with the OpenAI model, the performance scores are 75, 100, and 75 at temperatures 0, 0.2, and	×	0.04
For the Weights & Biases component with the WizardCoder model, the performance scores are 10, 30, and 50 at temperatures	×	0.08
For the PyTorch Lightning component with the WizardCoder model, the performance scores are 0, 0, and 50 at temperatures	×	0.05
Model Registration using MLflow requires initializing a new run using mlflow.start_run().	×	0.04

References

- http://arxiv.org/abs/2403.03788v1
- http://arxiv.org/abs/2405.06835v1
- http://arxiv.org/abs/2306.08568v2