# LF InfiniEdge AI Workstream#2: data and algorithms

- Open-source community with open-ended goals
- Edge use cases
- Exploring the boundary: LLMs + data and algorithms
  - Complex RAG (KDD 2024): Accuracy 20% ⇒ 70+% (with model fine tune and data preparation)
  - AI competitive coding (Neurips 2024): Accuracy 25% ⇒ 45% (Reflections) ⇒ 90% (Human Hints)

https://lf-edge.atlassian.net/wiki/spaces/IA/pages

EDA converts a user's *own* **data into** a *local,* on-demand **agent** service at the **edge**

*Data-on-prem, Code-on-the-Fly*

# Vibe Coding ⇒ Autonomous Coding

**Andrej Karpathy**
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are

## Vibe coding

Article Talk

From Wikipedia, the free encyclopedia

**Vibe coding** (also ***vibecoding***) is an AI-dependent programming technique where a person describes a problem in a few sentences as a prompt to a large language model (LLM) tuned for coding. The LLM generates software, shifting the programmer's role from manual coding to guiding, testing, and refining the AI-generated source code.[1][2][3] Vibe coding is claimed by its advocates to allow even amateur programmers to produce software without the extensive training and skills required for software engineering.[4] The term was introduced by Andrej Karpathy in February 2025[5][2][4][1] and listed in the Merriam-Webster Dictionary the following month as a "slang & trending" noun.[6]

### Definition [ edit ]

Computer scientist Andrej Karpathy, a co-founder of OpenAI and former AI leader at Tesla, introduced the term *vibe coding* in February 2025.[5][2][4][1] The concept refers to a coding approach that relies on LLMs, allowing programmers to generate working code by providing natural language descriptions rather than manually writing it.[1] Karpathy described his approach

SWE-bench Lite | SWE-bench Multimodal | SWE-bench Verified

**News**

- [10/2024] Introducing **SWE-bench Multimodal**! Can AI systems "see" bugs and fix them? 👀 [Link]
- [08/2024] SWE-bench x OpenAI = **SWE-bench Verified**, a human-validated subset of 500 problems reviewed by software engineers! [Report]
- [06/2024] We've **Docker**-ized SWE-bench for easier, containerized, reproducible evaluation. [Report]
- [03/2024] Check out our latest work, **SWE-agent**, which achieves a 12.47% resolve rate on SWE-bench! [Link]
- [03/2024] We've released **SWE-bench Lite**! Running all of SWE-bench can take time. This subset makes it easier! [Report]

**Leaderboard**

Lite | Verified | Full | Multimodal

| Model | % Resolved | Org | Date | Logs | Trajs | Site |
|---|---|---|---|---|---|---|
| Isoform | 55.00 | | 2025-01-14 | ✓ | ✓ | |
| Blackbox AI Agent | 49.00 | - | 2024-12-20 | ✓ | ✓ | |
| Gru(2024-12-08) | 48.67 | | 2024-12-08 | - | - | |
| Globant Code Fixer Agent | 48.33 | | 2024-11-27 | ✓ | ✓ | |
| devlo | 47.33 | | 2024-11-22 | ✓ | ✓ | |
| Kodu-v1 + Claude-3.5 Sonnet (20241022) | 44.67 | | 2024-12-07 | ✓ | ✓ | |
| ✅ OpenHands + CodeAct v2.1 (claude-3-5-sonnet-20241022) | 41.67 | | 2024-10-25 | ✓ | ✓ | |
| PatchKitty-0.9 + Claude-3.5 Sonnet (20241022) | 41.33 | | 2024-12-20 | ✓ | ✓ | - |
| Composio SWE-Kit (2024-10-30) | 41.00 | | 2024-10-30 | ✓ | ✓ | |
| Agentless-1.5 + Claude-3.5 Sonnet (20241022) | 40.67 | | 2024-12-02 | ✓ | ✓ | |
| OpenCSG Starship Agentic Coder + GPT 4 (0806) | 39.67 | - | 2025-01-13 | ✓ | ✓ | |
| Bytedance MarsCode Agent | 39.33 | | 2024-09-12 | ✓ | ✓ | |
| ✅ Moatless Tools + Claude 3.5 Sonnet (20241022) | 39.00 | - | 2025-01-14 | ✓ | ✓ | |
| ✅ Moatless Tools + Claude 3.5 Sonnet (20241022) | 38.33 | - | 2024-11-17 | ✓ | ✓ | |
| Honeycomb | 38.33 | | 2024-08-20 | ✓ | ✓ | |

# Agent: Developer or Consumer? ⇒ DevSumer?


- Developer + PM ⇒ App ⇒ Consumers
- DevSumer ⇒ Personalized App

# Edge Scenario

**Local / Domain Knowledge**

- E.g. “this database holds my latest inventory list”

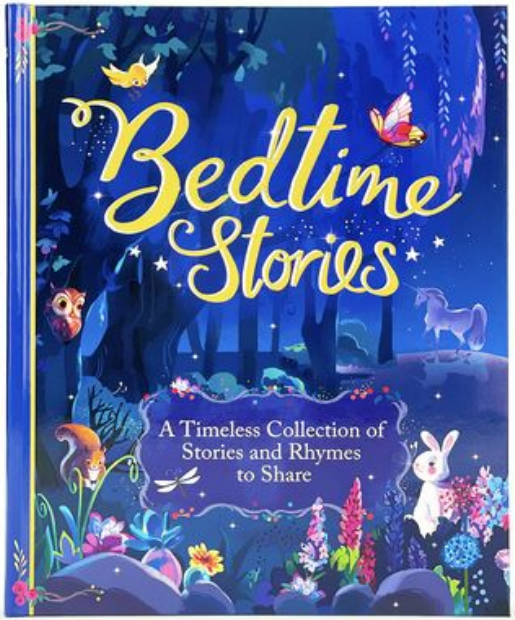
**User intention / Query history**

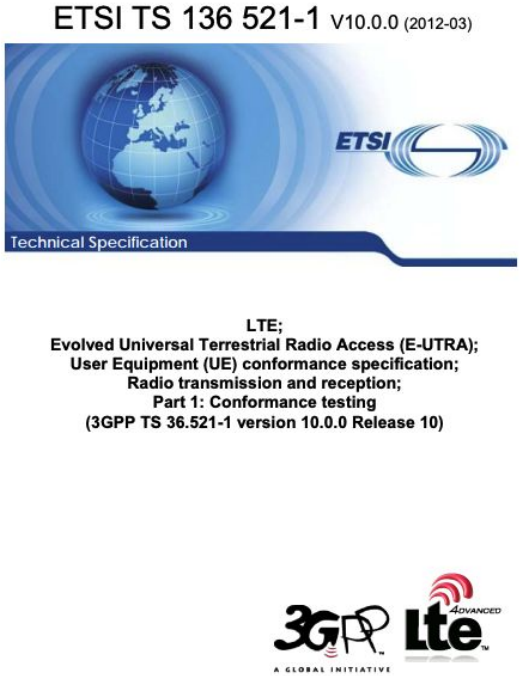
- E.g. List me the total number of SKUs in Product A

How can we infuse the above information into LLMs to better handle the task?

Generate the code (function / tool) for automation

# Examples




https://github.com/lfedgeai/eda/sandbox/data

Looking for developers, participants and contributors (Contact: qitang2023@gmail.com)