

# Toward Responsible AI in High-Stakes Domains: A Dataset for Building Static Analysis with LLMs in Structural Engineering

Carlos Avila <sup>1,\*</sup>, Daniel Ilbay <sup>2</sup>, Paola Tapia <sup>2</sup> and David Rivera <sup>2,\*</sup>

<sup>1</sup> Grupo de Investigación de Energía, Minas y Agua (GIEMA), Facultad de Ciencias, Ingeniería y Construcción, Universidad UTE, Quito 170527, Ecuador

<sup>2</sup> Facultad de Ciencias, Ingeniería y Construcción, Ingeniería Civil, Universidad UTE, Quito 170527, Ecuador; daniel.ilbay@ute.edu.ec (D.I.); paola.tapia@ute.edu.ec (P.T.)

\* Correspondence: carlos.avila@ute.edu.ec (C.A.); edgar.rivera@ute.edu.ec (D.R.)

## Abstract

Modern engineering increasingly operates within socio-technical networks, such as the interdependence of energy grids, transport systems, and building codes, where decisions must be reliable and transparent. Large language models (LLMs) such as GPT promise efficiency by interpreting domain-specific queries and generating outputs, yet their predictive nature can introduce biases or fabricated values—risks that are unacceptable in structural engineering, where safety and compliance are paramount. This work presents a dataset that embeds generative AI into validated computational workflows through the Model Context Protocol (MCP). MCP enables API-based integration between ChatGPT (GPT-4o) and numerical solvers by converting natural-language prompts into structured solver commands. This creates context-aware exchanges—for example, transforming a query on seismic drift limits into an OpenSees analysis—whose results are benchmarked against manually generated ETABS models. This architecture ensures traceability, reproducibility, and alignment with seismic design standards. The dataset contains prompts, GPT outputs, solver-based analyses, and comparative error metrics for four reinforced concrete frame models designed under Ecuadorian (NEC-15) and U.S. (ASCE 7-22) codes. The end-to-end runtime for these scenarios, including LLM prompting, MCP orchestration, and solver execution, ranged between 6 and 12 s, demonstrating feasibility for design and verification workflows. Beyond providing records, the dataset establishes a reproducible methodology for integrating LLMs into engineering practice, with three goals: enabling independent verification, fostering collaboration across AI and civil engineering, and setting benchmarks for responsible AI use in high-stakes domains.



Academic Editor: Bin Jiang

Received: 1 September 2025

Revised: 10 October 2025

Accepted: 22 October 2025

Published: 24 October 2025

**Citation:** Avila, C.; Ilbay, D.; Tapia, P.; Rivera, D. Toward Responsible AI in High-Stakes Domains: A Dataset for Building Static Analysis with LLMs in Structural Engineering. *Data* **2025**, *10*, 169. <https://doi.org/10.3390/data10110169>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Dataset:** 10.17632/gh9sbjzz5z.2

**Dataset License:** CC BY 4.0

**Keywords:** generative AI-assisted structural analysis; model context protocol; human-machine interaction; computational efficiency; LLM-FEM integration

## 1. Summary

This dataset accompanies the study “Human-AI Teaming in Structural Analysis: A Model Context Protocol Approach for Explainable and Accurate Generative AI” [1]. Engineering today operates in a landscape where complexity is no longer the exception but the rule. Energy grids, transport networks, and construction projects intertwine with social, economic, and

environmental systems, forming intricate socio-technical webs [2–5]. Within these environments, decision-making must not only be accurate but also transparent and reproducible, as even small misjudgements can cascade into consequences for safety, sustainability, and public trust. Against this backdrop, artificial intelligence (AI) has emerged as a transformative force, offering the ability to process vast datasets and generate insights at a speed unimaginable just a decade ago [6–10].

Among recent advances, large language models (LLMs) such as GPT have captured attention for their capacity to generate solutions, interpret technical queries, and adapt flexibly across domains with little additional training [8,11]. Their appeal lies in accessibility: engineers can pose complex questions in natural language and receive structured outputs almost instantly. Yet this promise carries an underexplored risk. By design, LLMs generate statistical predictions rather than verifiable computations [12–14]. What reads as a precise answer may, in fact, be a plausible illusion—embedding bias, misapplied logic, or entirely fabricated values. In structural engineering, where lives depend on fidelity to physical laws and regulatory codes, these hallucinations are more than an inconvenience; they are a critical barrier to safe adoption.

Addressing this tension requires moving beyond unbounded generative outputs toward context-aware frameworks that anchor AI reasoning within validated workflows. The present study employs the Model Context Protocol (MCP), a general-purpose standard that enables large language models to interact with external solvers through structured API communication [15]. MCP dynamically manages tool discovery, data exchange, and orchestration, ensuring that model prompts are translated into auditable solver inputs and outputs [15–18]. By connecting generative models directly with numerical engines such as OpenSees, the framework reduces hallucinations, enhances traceability, and ensures compliance with seismic-resistant design standards [19,20].

The dataset described herein illustrates this approach in practice. It comprises technical prompts, raw GPT outputs, validated numerical analyses conducted in OpenSeesPy (3.7.1, PEER) and ETABS (20.3.0, Computers and Structures, CSI), and comparative metrics across four three-dimensional reinforced concrete frames designed in accordance with the Ecuadorian Construction Standard NEC-15 [19] and the U.S. ASCE 7-22 [20]. Four modelling groups were evaluated: (i) GPT, using direct LLM prompting; (ii) GPT+MCP, integrating generative interaction with solver-based computation; and two benchmark groups, (iii) OpenSees and (iv) ETABS, developed manually for validation. The dataset reports inter-storey drifts in X and Y, maximum displacements (m), base shear (kN), and fundamental periods (s), with relative error analyses providing the basis for comparison. Beyond recording outputs, the dataset demonstrates a reproducible methodology for embedding LLMs into structural analysis workflows while preserving computational fidelity.

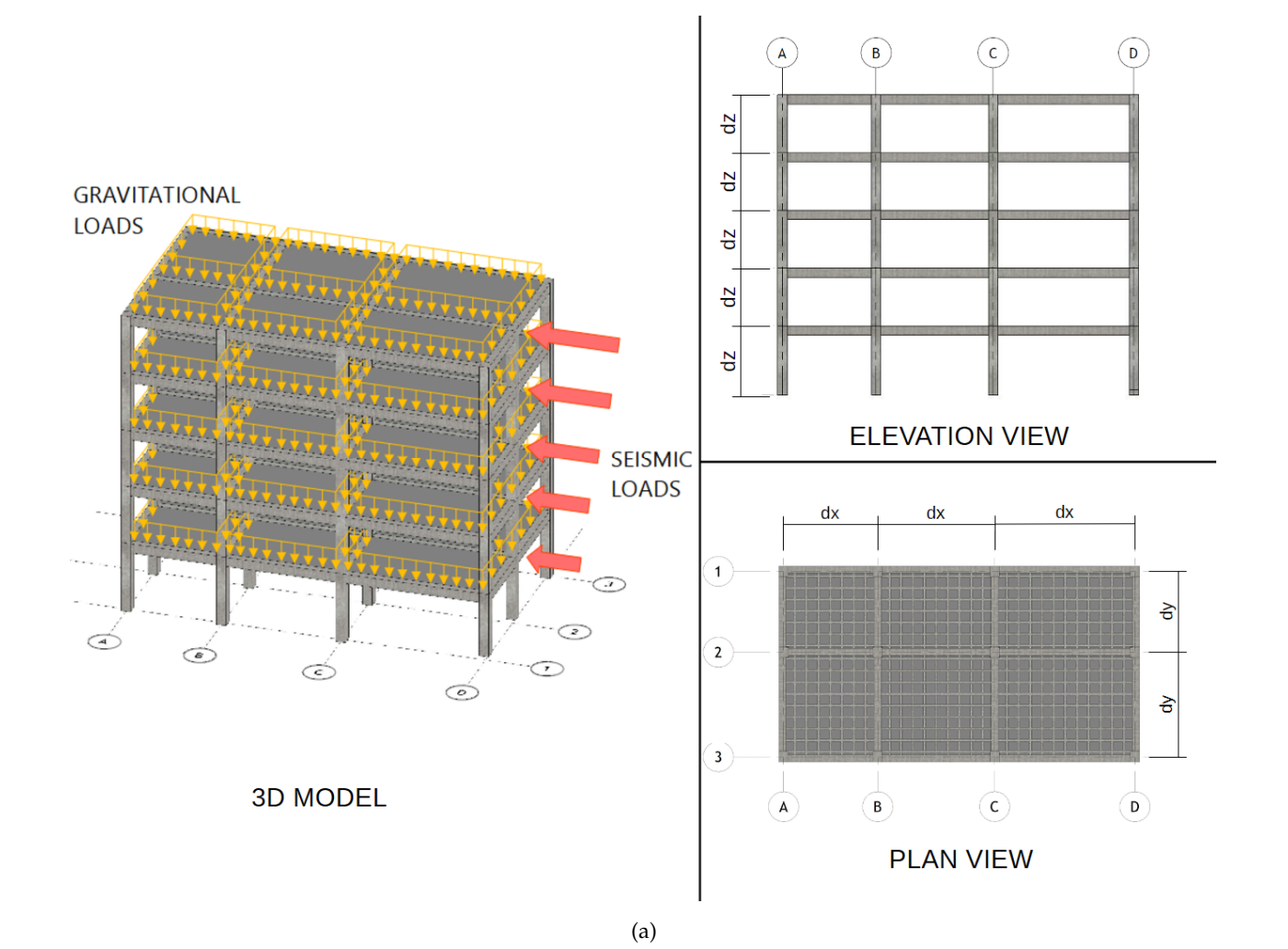
This work also lays a foundation for broader exploration. Current extensions of the MCP framework investigate domains such as energy modelling, HVAC analysis, and sustainability-driven optimisation, with parallel applications emerging in BIM-linked generative workflows [11,21,22]. By making the dataset publicly available, the study advances three objectives: (i) to strengthen reproducibility by enabling independent verification, (ii) to foster interdisciplinary collaboration between AI researchers, civil engineers, and data scientists, and (iii) to establish benchmarks for safe, context-aware AI integration in high-stakes engineering. Ultimately, this dataset highlights both the promise and the limits of generative AI in technical fields. It contributes not only to ongoing debates on explainability and trust in AI, but also to the practical development of workflows where human expertise and machine intelligence operate as genuine partners in structural decision-making.

2. Data Description

2.1. Prompt Description of the Study Case

The following subsection provides a structured explanation of the study case prompt, outlining its components, order, and interpretation to ensure clarity and reproducibility of the analysis (Table 1b).

**Table 1.** Structural Analysis Case Study: (a) Three-dimensional model implemented in OpenSees and ETABS, and (b) natural language specification expressed in MIDI-formatted prompt.



```
prompt = (  
  "Context:"  
  "You are an expert in structural analysis using natural language and numerical simulation with OpenSeesPy."  
  "The implemented system is capable of interpreting technical prompts and generating automated structural simulations"  
  "based on international seismic-resistant design standards."  
  "Instructions:"  
  "Analyse a three-dimensional reinforced concrete frame, verify code compliance for inter-storey drift,"  
  "and apply structural optimisation if necessary."  
  "Details:"  
  "The modulus of elasticity for concrete is 21,458,890.83 kN/m2."  
  "The structural system has spans of 4.0 and 4.0 m in the X direction, and spans of 4.0 and 4.0 m in the Y direction."  
  "The structure has 2 stories, with storey heights of: 3.0 and 3.0 m respectively."  
  "Beams have a cross-sectional dimension of 0.25 × 0.30 m, and columns are 0.30 × 0.30 m."
```

Table 1. Cont.

"Cracking factors are 0.7 for beams and 0.8 for columns."  
 "Dead load is 4.9 kN/m<sup>2</sup> and live load is 1.9 kN/m<sup>2</sup>."  
 "The weight coefficients are: 1.0 for dead load, 0.15 for live load, 0.1488 for base shear coefficient,"  
 "1.0 for vertical distribution of base shear, 0.05 for accidental torsion,"  
 "and a drift amplification factor of 6.0 is applied to estimate inelastic drift. The maximum allowable drift is 0.02."  
 "Tasks:"  
 "1. Perform linear static seismic analysis using the equivalent lateral force method with OpenSeesPy."  
 "2. Compute maximum displacements and storey drifts per level and direction (X and Y)."  
 "3. Perform strict numerical validation:"  
 " - Iterate through all obtained inelastic drift values."  
 " - For each value, compare it against the allowable maximum (0.02)."  
 " - If \*at least one value\* exceeds 0.02, \*you must not state that all values are compliant\*."  
 " - Report precisely: storey number, direction (X or Y), and the drift value that exceeds the limit."  
 " - Only if \*all drifts\* are  $\leq 0.02$ , the code compliance can be confirmed."  
 " - Present results in tabular format and be rigorous with numerical precision."  
 "4. Also determine floor-by-floor shear forces and vibration modes."  
 "5. Structural optimisation:"  
 " - If any drift exceeds the limit, propose a structural optimisation based on displacements,"  
 " storey drifts, shear forces, and vibration modes."  
 " - Generate 10 alternatives by modifying material properties and section dimensions."  
 " - Evaluate drift for each alternative and present the comparison in tabular format."  
 " - Highlight the configurations that meet code requirements and provide better structural efficiency."  
 "Intent:"  
 "Generate an automated technical report, including detailed structural analysis, code validation,"  
 "and optimisation in case of non-compliance. The output must be expressed in technical language and clear tables,"  
 "suitable for professional and academic environments."

(b)

Note: In panel (a), dx and dy denote the span lengths in the X and Y directions, respectively; their values vary according to the specific study case. The asterisk (\*) is used as markup to highlight terms and clarify user requirements, not as a mathematical or syntactic operator.

### 2.1.1. Context

The first block establishes the role of the system. It defines that the model must act as an expert in structural analysis, using both natural language and numerical simulation with OpenSeesPy. It also specifies that the analyses must follow international seismic-resistant design standards.

- Purpose: Provide the professional and technical framework within which all subsequent instructions must be interpreted.

### 2.1.2. Instructions

This section directs the type of analysis to be performed. The orders are as follows:

- Analyse a three-dimensional reinforced concrete frame.
- Verify code compliance for inter-storey drift.
- Apply structural optimisation if necessary.
- Purpose: Define the general workflow of the analysis.

### 2.1.3. Details

This block provides numerical input data required for analysis. The parameters are organised as structured values:

- Material properties: modulus of elasticity.
- Geometry: spans in X and Y directions; storey heights.
- Cross-sections: beam and column dimensions.
- Cracking factors: beams (0.7) and columns (0.8).

- Loads: dead load (4.9 kN/m<sup>2</sup>) and live load (1.9 kN/m<sup>2</sup>).
- Coefficients: load factors, base shear coefficient, torsion factor, drift amplification, and maximum allowable drift.
- Purpose: These values serve as tabular input data to be read directly by the solver. Each line corresponds to a parameter category, and their interpretation is straightforward (e.g., geometric dimensions in meters and load intensities in kN/m<sup>2</sup>).

#### 2.1.4. Tasks

This section enumerates the ordered computational tasks:

##### Task 1—Linear Static Seismic Analysis

Perform seismic analysis with the equivalent lateral force method using OpenSeesPy.

##### Task 2—Displacements and Drifts

Compute maximum displacements and storey drifts for each level in both directions (X and Y).

##### Task 3—Strict Numerical Validation

- Iterate through all inelastic drift values.
- Compare against maximum allowable drift (0.02).
- If one or more values exceed the limit, report the storey number, direction, and drift value.
- Only if all drifts are  $\leq 0.02$  may compliance be confirmed.
- Present results in tabular format with numerical precision.

##### Task 4—Shear Forces and Vibration Modes

Determine floor-by-floor shear forces and vibration modes.

##### Task 5—Structural Optimisation

- If non-compliance occurs, generate 10 alternatives by modifying materials and section dimensions.
- Re-evaluate drifts for each configuration.
- Compare alternatives in tabular form.
- Highlight compliant and efficient options.

#### 2.1.5. Intent

The final block specifies the expected output style:

- Generate an automated technical report.
- Include detailed structural analysis, code validation, and optimisation proposals when required.
- Use technical language with clear tables.
- Ensure suitability for professional and academic environments.

### 2.2. Dataset Significance

The dataset generated in this study is organised around four primary indicators of seismic performance—storey drift, maximum displacement, base shear, and building period—evaluated across four structural cases (A–D) using standalone GPT, GPT+MCP, OpenSees, and ETABS. The four study cases were specifically chosen to evaluate the methodology across a spectrum of design challenges common in structural practice. Case A establishes a baseline with a simple, symmetric low-rise frame. Case B introduces geometric asymmetry to test the system's handling of torsional effects. Cases C and D scale the problem to a mid-rise structure, again examining both symmetric and asymmetric configurations to assess the robustness and scalability of the AI-assisted workflow. Table 2 provides the details of the modelling setup for Study Case A, including geometry, loading

conditions, and material properties. To broaden the dataset's applicability and ensure a more comprehensive assessment of the GPT+MCP workflow, further study cases were included and are summarised in Table S1. These additional configurations introduce controlled variations in geometry, material stiffness, and seismic demand, allowing users to evaluate the stability and generalisability of the proposed framework under diverse structural conditions. The data are presented in tabular and graphical formats, with consistent units: meters (m) for displacements, dimensionless ratios for storey drift, kilonewtons (kN) for base shear, and seconds (s) for the building period. This structure ensures comparability between approaches and facilitates interpretation against seismic code requirements.

**Table 2.** Geometric, mechanical, and loading parameters defining the 4 study cases.

| Category           | Parameter                                     | Case A             | Case B             | Case C              | Case D              |
|--------------------|-----------------------------------------------|--------------------|--------------------|---------------------|---------------------|
| Geometry           | No. of Stories                                | 2                  | 3                  | 5                   | 5                   |
|                    | Storey Heights (m)                            | 3.0–3.0            | 3.0–3.0–3.0        | 3.5–2.5–2.5–2.5–2.5 | 3.0–2.5–2.5–2.5–2.5 |
|                    | Spans in X (m)                                | 4.0–4.0            | 4.0–5.0–4.0        | 4.0–4.0–4.0         | 4.0–5.0–6.0         |
|                    | Spans in Y (m)                                | 4.0–4.0            | 3.5–4.5            | 4.0–4.0–4.0         | 3.5–4.5             |
| Sections           | Geometry Type                                 | Symmetric          | Asymmetric         | Symmetric           | Asymmetric          |
|                    | Beam Cross-Section (m)                        | $0.25 \times 0.30$ | $0.30 \times 0.30$ | $0.30 \times 0.40$  | $0.30 \times 0.40$  |
|                    | Column Cross-Section (m)                      | $0.30 \times 0.30$ | $0.35 \times 0.35$ | $0.40 \times 0.40$  | $0.40 \times 0.45$  |
| Material           | Cracking Factor (Beams)                       |                    |                    | 0.7                 |                     |
|                    | Cracking Factor (Columns)                     |                    |                    | 0.8                 |                     |
|                    | Concrete Young's Modulus (kN/m <sup>2</sup> ) |                    |                    | 21,458,890.83       |                     |
| Loads              | Dead Load (kN/m <sup>2</sup> )                |                    |                    | 4.9                 |                     |
|                    | Live Load (kN/m <sup>2</sup> )                |                    |                    | 1.9                 |                     |
|                    | Dead Load Coefficient                         |                    |                    | 1                   |                     |
|                    | Live Load Coefficient                         |                    |                    | 0.15                |                     |
|                    | Base Shear Coefficient                        |                    |                    | 0.1488              |                     |
| Seismic Parameters | Vertical Distribution Coefficient             |                    |                    | 1                   |                     |
|                    | Accidental Torsion Coefficient                |                    |                    | 0.05                |                     |
|                    | Drift Amplification Factor                    |                    |                    | 6                   |                     |
|                    | Maximum Allowable Drift                       |                    |                    | 0.02                |                     |

Note: table interpretation of the structured prompt.

- **Storey Drift:** Storey drift values are reported for each storey in both the X and Y directions (Table 3). These tables allow readers to observe the vertical distribution of drift across cases and computational methods. Storey drift quantifies the relative displacement between consecutive levels and is a key parameter for NEC-15 compliance, which establishes a 2% upper limit. *Reading guide: Table 3 displays raw drift values per storey and direction, enabling direct verification of inter-storey deformation patterns.*

**Table 3.** Storey drift in the X direction computed for the study case A.

| Inter-Storey Drift X |        |       |         |          |       |
|----------------------|--------|-------|---------|----------|-------|
| Case                 | Storey | GPT   | GPT+MCP | OpenSees | ETABS |
| A                    | 2      | 0.009 | 0.013   | 0.013    | 0.013 |
|                      | 1      | 0.007 | 0.012   | 0.012    | 0.012 |

- **Maximum Displacement:** The maximum storey displacements in the X and Y directions summarised numerically in Table 4. This dataset provides insight into global deformation profiles, which are essential for evaluating the likelihood of structural interaction with neighboring buildings. *Reading guide: Table 4 reports the corresponding numerical values of the displacement in meters for each computational method.*

**Table 4.** Maximum inter-storey displacement in the X (m).

| Case | GPT   | GPT+MCP | OpenSees | ETABS   |
|------|-------|---------|----------|---------|
| A    | 0.047 | 0.01264 | 0.01264  | 0.01282 |

- **Base Shear:** Table 5 presents the base shear (kN) values for the studied cases. These results quantify the seismic demand transmitted to the foundation and reflect the combined influence of structural weight and stiffness. *Reading guide: Table 5 provides total base shear per case and method, facilitating cross-model comparison.*

**Table 5.** Base shear (kN) in study case A.

| Case | GPT   | GPT+MCP | OpenSees | ETABS |
|------|-------|---------|----------|-------|
| A    | 10.09 | 13.969  | 13.969   | 13.97 |

- **Building Period.** The fundamental period of vibration for each case is shown in Table 6. As an indirect measure of stiffness, the building period is critical for understanding overall structural dynamics, where shorter periods generally correspond to reduced displacements. *Reading guide: Table 6 presents the fundamental period of vibration for Study Case A across the different modelling scenarios (GPT, GPT+MCP, OpenSees, and ETABS). This comparison highlights the consistency of solver-based methods relative to ETABS, while also showing the divergence of GPT-only outputs. The fundamental period serves as an indicator of global structural stiffness, where shorter values generally reflect stiffer dynamic behaviour.*

**Table 6.** Building period (s) in study case A.

| Case | GPT  | GPT+MCP | OpenSees | ETABS |
|------|------|---------|----------|-------|
| A    | 0.38 | 0.485   | 0.485    | 0.489 |

Taken together, these datasets form the empirical basis of the study's argument. The definition of consistent study groups supports validation across computational approaches, while the structured results are presented through storey drift, displacement, base shear, and period. These indicators collectively establish compliance with NEC-15 drift limits, with GPT+MCP offering corrective alternatives when design parameters exceed code requirements.

The dataset demonstrates that the GPT+MCP framework, integrated with OpenSees via the MCP protocol, produces results consistent with conventional structural analysis software such as ETABS, while markedly improving computational efficiency—reducing run times from approximately 12–15 min (manual ETABS integration) to only 6–12 s. The dataset therefore not only enables verification of outputs but also provides a foundation for future studies exploring new design alternatives or extending AI-assisted methodologies to other structural typologies.

### 2.3. Relative Error as a Measure of Accuracy

To complement the raw structural response results, the dataset also contains derived values of relative error, calculated with respect to ETABS benchmarks. The metric was computed using Equation (1) [1]:

$$\text{Relative Error} = \frac{|X_{\text{model}} - X_{\text{ETABS}}|}{X_{\text{ETABS}}} \times 100\% \quad (1)$$

where X represents one of the evaluated structural response parameters: storey drift, maximum displacement, base shear, or period. This formulation normalises deviations across methods, making results directly comparable regardless of units or magnitude.

The dataset includes separate relative error tables for each parameter, as well as summary tables reporting the maximum relative errors across cases A–D. For instance, Table 7 presents the relative error for base shear. Each column corresponds to a computational method (GPT, GPT+MCP, and OpenSees), and each row indicates a structural case. An additional row reports the standard deviation (SD) of the maximum error across cases, which reflects the stability of the method. The values are expressed in percentages (%).

**Table 7.** Relative error (%) with respect to ETABS in the evaluation of max displacement in the X and Y directions.

| Case | GPT     |         | GPT+MCP |       | OpenSees |       |
|------|---------|---------|---------|-------|----------|-------|
|      | X       | Y       | X       | Y     | X        | Y     |
| A    | 266.529 | 235.335 | 1.427   | 1.427 | 1.427    | 1.427 |

**How to read the data.** The relative error column for each method indicates its proportional deviation from ETABS. Values closer to zero denote higher accuracy and stronger agreement with the benchmark. The inclusion of SD allows users to evaluate not only the accuracy in individual cases but also the variability across different structural configurations. For example, the GPT-only method exhibits high relative errors (230–270%) with large dispersion, while GPT+MCP and OpenSees maintain relative errors consistently close to zero (<1.427%) with negligible variability.

**Relevance of the metric.** The relative error dataset enables three main uses the following:

- *Benchmarking performance:* It provides a direct comparison of novel AI-assisted methods (GPT and GPT+MCP) against a widely validated standard (ETABS).
- *Cross-parameter evaluation:* Since relative error is unitless, it permits consistent assessment across storey drift (%), displacements (m), base shear (kN), and period (s).
- *Reproducibility and extension:* Researchers can employ the provided error tables to reproduce the evaluation, extend the analysis to new structural typologies, or integrate the metric into broader model validation frameworks.

By including relative error in addition to raw results, the dataset provides a transparent and standardised measure of accuracy. This reinforces the robustness of the GPT+MCP approach, which is shown to achieve performance commensurable with established engineering tools. These relative error metrics therefore provide the quantitative basis for evaluating the reliability of AI-assisted structural analysis in high-stakes domains. For extended interpretation of these results and their implications in engineering practice, readers are referred to the full dataset (DOI: 10.17632/gh9sbjzz5z.2) and the companion article published in Buildings (DOI: 10.3390/buildings15173190), where the outcomes and practical significance are analysed in detail.

#### 2.4. Dataset Statistics

The dataset comprises four study cases, each defined by a single structured CIDI-style prompt (Table 1). For every case, analyses were performed under four computational groups: GPT, GPT+MCP, OpenSees, and ETABS. This structure yields a total of 16 analyses, with outputs systematically reported for storey drift, maximum displacement, base shear, and building period. By explicitly defining the number of prompts and analyses, the dataset provides a transparent scope that facilitates reproducibility and benchmarking.

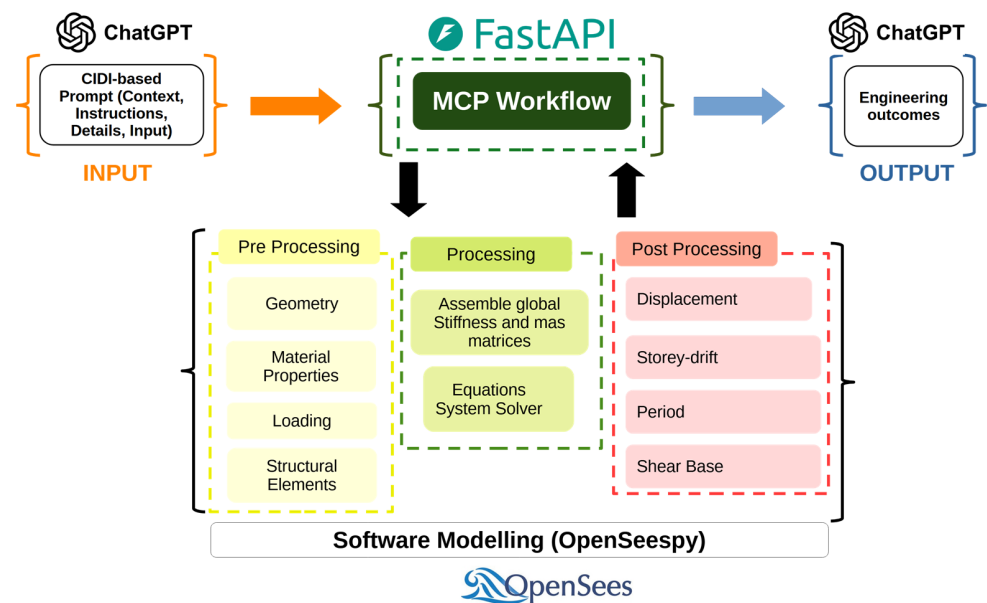
### 2.5. Potential for Reuse and Future Research

Beyond verifying our companion study [1], this dataset is a foundational resource for the research community. Its standardised prompts and validated benchmarks establish a testbed for evaluating new AI models and human-AI interaction protocols. It provides a template for future work, enabling researchers to build more comprehensive datasets by introducing greater complexity, such as different materials, advanced analysis types (e.g., nonlinear procedures), or potential structural optimisation (Task 5). The dataset also serves as a practical tool for educational and training purposes, facilitating the exploration of LLMs in a controlled environment. By supporting these avenues, this work is designed to be a dynamic contribution that encourages further innovation at the intersection of artificial intelligence and structural engineering.

## 3. Methods

### 3.1. Architecture Workflow

The dataset was generated using a modular client–server workflow structured under the Model–Context Protocol (Figure 1). The architecture separates natural language reasoning from structural simulation, thereby ensuring transparency and reproducibility. Three functional layers were defined. The Client Layer processes user prompts through a large language model (GPT-4o) and translates them into structured JSON schemas specifying geometry, materials, and load conditions. The Server Layer, implemented with FastAPI (3.1.1), validates inputs, orchestrates tool invocation, and manages execution order. The External Application Layer integrates OpenSeesPy (3.8.x interface to OpenSees 3.7.1), which performs seismic analyses including inter-storey drift evaluation, shear force distribution, and modal response (Figure 1).



**Figure 1.** Workflow for dataset generation using the GPT+MCP pipeline.

### 3.2. Data Collection and Processing

Natural language prompts (CIDI-style specifications) were employed to generate multiple reinforced concrete frame models (Tables 1 and 2). Parameters included storey heights, spans, cross-sectional dimensions, and seismic coefficients. Each prompt was parsed into machine-readable inputs, which OpenSeesPy converted into three-dimensional analysis models. Structural outputs consisted of storey drifts, shear distributions, and

vibration modes. These raw outputs were compiled into structured JSON tables to ensure consistency and ease of reuse.

### 3.3. Validation and Curation

Validation was performed through two complementary strategies. First, syntactically distinct but semantically equivalent prompts were tested to verify that the system produced consistent model definitions. Second, results from GPT+MCP workflows were benchmarked against manually implemented OpenSees and ETABS (20.3.0, Computers and Structures, CSI) models. ETABS was selected as the benchmark for its reliability, global recognition in structural analysis, and the quality standards, i.e., ISO 9001:2015, maintained by CSI. Relative error analysis quantified deviations across platforms, with results showing near-identical outputs for GPT+MCP and manual OpenSees models. Curation procedures included schema enforcement, error logging, and systematic rejection of incomplete inputs.

### 3.4. Data Quality and Noise Control

Potential sources of noise included missing prompt parameters, incorrectly typed values, solver non-convergence, and infeasible geometries. A two-tier error-handling mechanism was implemented. Client-side validation ensured compliance with structured JSON schemas, while server-side monitoring detected solver errors or stability thresholds. Ambiguities and inconsistencies were explicitly logged, and outputs failing validation were excluded from the curated dataset. These mechanisms ensure high-quality, reproducible data suitable for downstream analysis.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/data10110169/s1>, Table S1: Additional Study Cases.

**Author Contributions:** Conceptualisation, C.A. and D.R.; formal analysis, C.A.; investigation, D.R. and P.T.; methodology, D.I. and P.T.; software, C.A. and D.I.; supervision, C.A. and D.R.; validation, D.I.; writing—original draft, D.I., D.R. and P.T.; writing—review and editing, C.A. and D.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets generated and analysed during the current study are publicly available from Mendeley Data at <https://doi.org/10.17632/gh9sbjzz5z.2> (accessed on 7 October 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|      |                                            |
|------|--------------------------------------------|
| AI   | Artificial Intelligence                    |
| LLM  | Large Language Model                       |
| GPT  | Generative Pre-trained Transformer         |
| MCP  | Model Context Protocol                     |
| BIM  | Building Information Modelling             |
| HVAC | Heating, Ventilation, and Air Conditioning |

|            |                                                           |
|------------|-----------------------------------------------------------|
| NEC-15     | Norma Ecuatoriana de la Construcción 2015                 |
| ASCE 7-22  | American Society of Civil Engineers Standard 7-2022       |
| ETABS      | Extended Three-dimensional Analysis of Building Systems   |
| OpenSees   | Open System for Earthquake Engineering Simulation         |
| OpenSeesPy | Python interface to OpenSees                              |
| JSON       | JavaScript Object Notation                                |
| API        | Application Programming Interface                         |
| FastAPI    | Fast Application Programming Interface (Python framework) |
| CIDI       | Context–Instruction–Details–Intent                        |

## References

1. Avila, C.; Ilbay, D.; Rivera, D. Human–AI Teaming in Structural Analysis: A Model Context Protocol Approach for Explainable and Accurate Generative AI. *Buildings* **2025**, *15*, 3190. [\[CrossRef\]](#)
2. Garza Morales, G.A.; Nizamis, K.; Bonnema, G.M. Engineering complexity beyond the surface: Discerning the viewpoints, the drivers, and the challenges. *Res. Eng. Des.* **2023**, *34*, 367–400. [\[CrossRef\]](#)
3. Morales, G.A.G.; Nizamis, K.; Bonnema, G.M. Why is there complexity in engineering? A scoping review on complexity origins. In Proceedings of the IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 17–20 April 2023; pp. 1–8. [\[CrossRef\]](#)
4. Suh, N.P. Complexity in Engineering. *CIRP Ann.* **2005**, *54*, 46–63. [\[CrossRef\]](#)
5. Oladele, A., Jr.; Ibrahim, A. Artificial Intelligence For Systems Engineering Complexity: A Review On The Use Of Ai And Machine Learning Algorithms. *Comput. Sci. IT Res. J.* **2024**, *5*, 787–808. [\[CrossRef\]](#)
6. Liang, H.; Kalaleh, M.T.; Mei, Q. Integrating Large Language Models for Automated Structural Analysis. *arXiv* **2025**. [\[CrossRef\]](#)
7. Cha, Y.-J.; Ali, R.; Lewis, J.; Büyüköztürk, O. Deep learning-based structural health monitoring. *Autom. Constr.* **2024**, *161*, 105328. [\[CrossRef\]](#)
8. Zhang, L.; Le, B.; Akhtar, N.; Lam, S.-K.; Ngo, T. Large Language Models for Computer-Aided Design: A Survey. *ACM Comput. Surv.* **2025**, *37*, 31. [\[CrossRef\]](#)
9. Salehi, H.; Burgueño, R. Emerging artificial intelligence methods in structural engineering. *Eng. Struct.* **2018**, *171*, 170–189. [\[CrossRef\]](#)
10. Cáceres, M.; Avila, C.; Rivera, E. Thermodynamics-Informed Neural Networks for the Design of Solar Collectors: An Application on Water Heating in the Highland Areas of the Andes. *Energies* **2024**, *17*, 4978. [\[CrossRef\]](#)
11. Lu, J.; Tian, X.; Zhang, C.; Zhao, Y.; Zhang, J.; Zhang, W.; Feng, C.; He, J.; Wang, J.; He, F. Evaluation of large language models (LLMs) on the mastery of knowledge and skills in the heating, ventilation and air conditioning (HVAC) industry. *Energy Built Environ.* **2024**, *6*, 875–892. [\[CrossRef\]](#)
12. Yang, X.; Chen, B.; Tam, Y.-C. Arithmetic Reasoning with LLM: Prolog Generation & Permutation. In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, Mexico City, MX, USA, 16–21 June 2024.
13. Ismayilzada, M.; Paul, D.; Montariol, S.; Geva, M.; Bosselut, A. CRoW: Benchmarking Commonsense Reasoning in Real-World Tasks. Conference on Empirical Methods in Natural Language Processing. *arXiv* **2023**. [\[CrossRef\]](#)
14. Ghimire, P.; Kim, K.; Acharya, M. Opportunities and Challenges of Generative AI in Construction Industry: Focusing on Adoption of Text-Based Models. *Buildings* **2024**, *14*, 220. [\[CrossRef\]](#)
15. Anthropic. Introducing the Model Context Protocol 2024. Available online: <https://www.anthropic.com/news/model-context-protocol> (accessed on 17 June 2025).
16. Krishnan, N. Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications. *arXiv* **2025**, arXiv:2504.21030. [\[CrossRef\]](#)
17. Hou, X.; Zhao, Y. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. *arXiv* **2025**. [\[CrossRef\]](#)
18. Ray, P.P.; Pratim, P.R. A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions. *techrXiv* **2025**. [\[CrossRef\]](#)
19. Norma Ecuatoriana de la Construcción—NEC: NEC-SE-MP. *Mampostería Estructural*; Camicon Miduvi: Quito, Ecuador, 2014.
20. ASCE/SEI 7-22; Minimum Design Loads and Associated Criteria for Buildings and Other Structures. American Society of Civil Engineers: Reston, VI, USA, 2022.

21. Jiang, G.; Chen, J. Efficient fine-tuning of large language models for automated building energy modeling in complex cases. *Autom. Constr.* **2025**, *175*, 106223. [[CrossRef](#)]
22. Jurišević, N.; Kowalik, R.; Gordić, D.; Novaković, A.; Vukašinović, V.; Rakić, N.; Nikolic, J.; Vukicevic, A. Large Language Models as Tools for Public Building Energy Management: An Assessment of Possibilities and Barriers. *Int. J. Qual. Res.* **2025**, *19*, 817–830. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.