

Data Descriptor

Not peer-reviewed version

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

[Carlos Fabian Avila Vega](#)*, [Daniel Jefferson Ibay Yupa](#), [Paola Vanessa Tapia Tapia](#),
[Edgar David Rivera Tapia](#)*

Posted Date: 2 September 2025

doi: 10.20944/preprints202509.0212.v1

Keywords: Generative AI-assisted Structural Analysis; Model Context Protocol; human-machine Interaction; Computational efficiency; LLM-FEM integration



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Data Descriptor

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

Carlos Avila ^{1,*}, Daniel Ilbay ², Paola Tapia ² and David Rivera ^{2,*}

- ¹ Grupo de Investigación de Energía, Minas y Agua (GIEMA), Facultad de Ciencias, Ingeniería y Construcción, Universidad UTE, Quito 170527, Ecuador
- ² Facultad de Ciencias, Ingeniería y Construcción, Ingeniería Civil, Universidad UTE, Quito 170527, Ecuador
- * Correspondence: carlos.avila@ute.edu.ec (C.A.); edgar.rivera@ute.edu.ec (D.R.)

Abstract

Modern engineering faces an unprecedented paradox: while our systems grow increasingly complex, the tools we use to design and evaluate them must remain both reliable and transparent. Decisions in energy, infrastructure, and construction no longer occur in isolation but within socio-technical networks shaped by emerging technologies and artificial intelligence (AI). Among these advances, large language models (LLMs) such as GPT have attracted attention for their ability to synthesize solutions, interpret domain-specific queries, and generate outputs with minimal fine-tuning. Yet beneath this promise lies a critical flaw—LLMs do not compute; they predict. Their reliance on statistical associations often leads to biases, logical missteps, or hallucinated values, shortcomings that become especially problematic when applied to structural engineering, where safety and compliance are non-negotiable. This tension sets the stage for the present work. The dataset introduced here responds to this gap by demonstrating how generative AI can be grounded within validated computational workflows. Through the Model Context Protocol (MCP), ChatGPT was connected to numerical solvers such as OpenSees and benchmarked against ETABS, ensuring traceability, reproducibility, and compliance with seismic design standards. The dataset comprises technical prompts, GPT outputs, verified numerical analyses, and comparative error metrics for four reinforced concrete frame models designed under Ecuadorian (NEC-15) and U.S. (ASCE 7-22) standards. Beyond a simple record, it exemplifies a reproducible methodology for embedding LLMs within structural engineering practice. By curating and releasing this dataset, the study pursues three goals: to strengthen reproducibility by enabling independent verification, to foster interdisciplinary collaboration across AI, civil engineering, and data science, and to establish benchmarks for context-aware AI integration in high-stakes domains. In doing so, it not only illustrates the promise of human–AI teaming but also highlights the limitations that must be addressed if generative models are to be responsibly embedded in engineering decision-making.

Dataset: 10.17632/gh9sbjzz5z.1

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 takes up this challenge by employing the Model Context Protocol (MCP), a framework that natively supports autonomous tool discovery and structured communication with external solvers [15–18]. Unlike orchestration tools such as LangChain [19], MCP allows generative models to interoperate directly with numerical engines such as OpenSees, ensuring that outputs are traceable, reproducible, and auditable. This integration not only reduces hallucinations but also enables compliance with established seismic-resistant design standards [20,21].

The dataset described here illustrates this approach in practice. It comprises technical prompts, raw GPT outputs, validated numerical analyses conducted in OpenSeesPy and ETABS, and comparative metrics across four three-dimensional reinforced concrete frames designed in accordance with the Ecuadorian Construction Standard NEC-15 [20] and the U.S. ASCE 7-22 [21]. 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 optimization, with parallel applications emerging in BIM-linked generative workflows [11,22,23]. 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)

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:

- Analyse a three-dimensional reinforced concrete frame.
- Verify code compliance for inter-story 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: beams and columns dimensions.
- Cracking factors: beams (0.7), columns (0.8).
- Loads: dead load (4.9 kN/m²), live load (1.9 kN/m²).
- Coefficients: load factors, base shear coefficient, torsion factor, drift amplification, 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, load intensities in kN/m²).

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 story 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 story number, direction, and drift value.
- Only if all drifts are ≤ 0.02 , compliance may 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

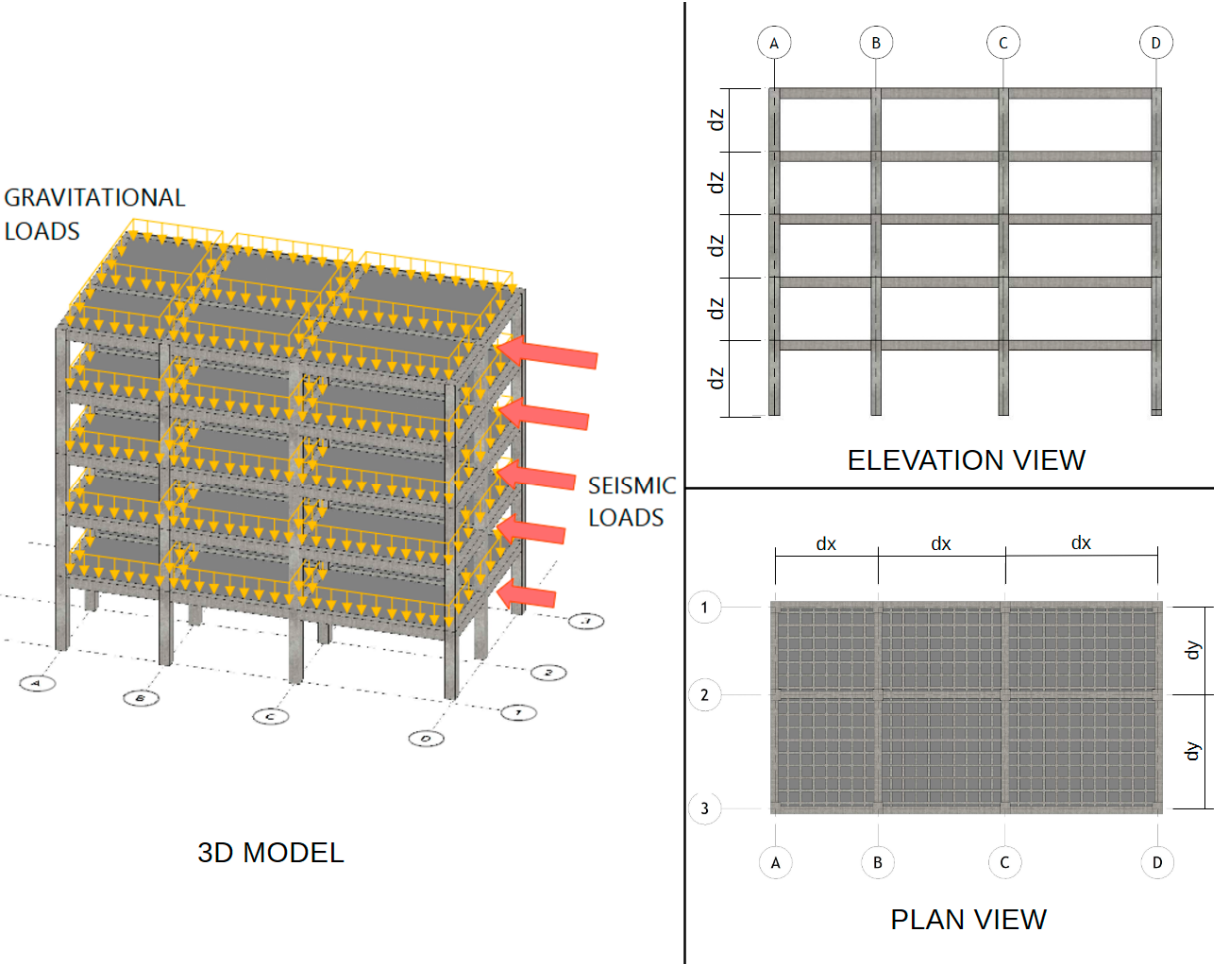
- If non-compliance occurs, generate 10 alternatives by modifying materials and section dimensions.
- Re-evaluate drifts for each configuration.
- Compare alternatives in tabular form.
- Highlight compliant and efficient options.

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.

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.



(a)

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-story drift, "

"and apply structural optimization 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 meters in the X direction, and spans of 4.0 and 4.0 meters in the Y direction. "

"The structure has 2 stories, with story heights of: 3.0 and 3.0 meters respectively. "

"Beams have a cross-sectional dimension of 0.25 x 0.30 meters, and columns are 0.30 x 0.30 meters. "

"Cracking factors are 0.7 for beams and 0.8 for columns. "

"Dead load is 4.9 Kn/m² and live load is 1.9 kN/m². "

"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 story 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: story number, direction (X or Y), and the drift value that exceeds the limit. "

" - Only if *all drifts* are ≤ 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 optimization:"

" - 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)

^{1.} 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.

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) (Table 2) using standalone GPT, GPT+MCP, OpenSees, and ETABS. 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 study case A.

Category	Parameter	Magnitude
Geometry	Number of Stories	2
	Story Heights	3.0, 3.0 m (from bottom to top)
	Spans in X Direction	4.0 m, 4.0 m
Sections	Spans in Y Direction	4.0 m, 4.0 m
	Beam Cross-Section	0.25 m × 0.30 m
	Column Cross-Section	0.30 m × 0.30 m
	Cracking Factor (Beams)	0.7
	Cracking Factor (Columns)	0.8
Material	Concrete Young’s Modulus	21,458,890.83 kN/m²
Loads	Dead Load	4.9 kN/m²
	Live Load	1.9 kN/m²
	Dead Load Weight Coefficient	1.0
	Live Load Weight Coefficient	0.15
Seismic Parameters	Base Shear Coefficient	0.1488
	Vertical Distribution Coefficient	1.0
	Accidental Torsion Coefficient	0.05
	Drift Amplification Factor (for Inelastic Drift)	6.0
	Maximum Allowable Drift	0.02

- ² 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 display 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-Story Drift X					
Case	Story	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 story 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 interstorey displacement in the X (m).

Storey	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: Figure 4 presents the variation of the fundamental period across study cases, while Table 11 provides the tabulated values per method.*

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 GPT+MCP, integrated with OpenSees through the MCP protocol, delivers results consistent with conventional software benchmarks, while reducing computation times. 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):

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

(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, 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.

GPT	GPT+MCP	OPENSEES
-----	---------	----------

Case	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:
- *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 standardized measure of accuracy. This reinforces the robustness of the GPT+MCP approach, which is shown to achieve performance commensurable with established engineering tools.

3. Methods

3.1. Architecture Workflow

The dataset was generated using a modular client–server workflow structured under the Model–Context Protocol. 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.

3.2. Data Collection and Processing

Natural language prompts (CIDI-style specifications) were employed to generate multiple reinforced concrete frame models (Table 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) models. 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.

3.5. Data Quality and Noise Control

Potential sources of noise included missing prompt parameters, ill-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.

Author Contributions: Conceptualisation, C.A. and D.R.; formal analysis, C.A.; investigation, D.R. and P.T.; methodology, D.I., 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 analyzed during the current study are publicly available in Mendeley Data at: <https://doi.org/10.17632/gh9sbjzz5z.1>.

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 Modeling
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 2025. <https://doi.org/10.20944/preprints202507.1968.v1>.
2. Garza Morales GA, Nizamis K, Bonnema GM. Engineering complexity beyond the surface: discerning the viewpoints, the drivers, and the challenges. Res Eng Des 2023;34:367–400. <https://doi.org/10.1007/S00163-023-00411-9/FIGURES/11>.
3. Morales GAG, Nizamis K, Bonnema GM. Why is there complexity in engineering? A scoping review on complexity origins. 2023 IEEE International Systems Conference (SysCon), IEEE; 2023, p. 1–8. <https://doi.org/10.1109/SysCon53073.2023.10131068>.
4. Suh NP. Complexity in Engineering. CIRP Annals 2005;54:46–63. [https://doi.org/10.1016/S0007-8506\(07\)60019-5](https://doi.org/10.1016/S0007-8506(07)60019-5).

5. Oladele Junior Adeyeye, Ibrahim Akanbi. ARTIFICIAL INTELLIGENCE FOR SYSTEMS ENGINEERING COMPLEXITY: A REVIEW ON THE USE OF AI AND MACHINE LEARNING ALGORITHMS. *Computer Science & IT Research Journal* 2024;5:787–808. <https://doi.org/10.51594/csitrj.v5i4.1026>.
6. Liang H, Kalaleh MT, Mei Q. Integrating Large Language Models for Automated Structural Analysis 2025. <https://doi.org/https://doi.org/10.48550/arXiv.2504.09754>.
7. Cha Y-J, Ali R, Lewis J, Büyüköztürk O. Deep learning-based structural health monitoring. *Autom Constr* 2024;161:105328. <https://doi.org/10.1016/j.autcon.2024.105328>.
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. <https://doi.org/XXXXXXX.XXXXXXX>.
9. Salehi H, Burgueño R. Emerging artificial intelligence methods in structural engineering. *Eng Struct* 2018;171:170–89. <https://doi.org/10.1016/j.engstruct.2018.05.084>.
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 (Basel)* 2024;17. <https://doi.org/10.3390/en17194978>.
11. Lu J, Tian X, Zhang C, Zhao Y, Zhang J, Zhang W, et al. Evaluation of large language models (LLMs) on the mastery of knowledge and skills in the heating, ventilation and air conditioning (HVAC) industry. *Energy and Built Environment* 2024. <https://doi.org/10.1016/j.ENBENV.2024.03.010>.
12. Yang X, Chen B, Tam Y-C. Arithmetic Reasoning with LLM: Prolog Generation & Permutation. 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Mexico: 2024.
13. Ismayilzada M, Paul D, Montariol S, Geva M, Bosselut A. CRow: Benchmarking Commonsense Reasoning in Real-World Tasks. *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. <https://doi.org/https://doi.org/10.48550/arXiv.2310.15239>.
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. <https://doi.org/10.3390/buildings14010220>.
15. Krishnan N. Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications 2025.
16. Hou X, Zhao Y. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions 2025;1. <https://doi.org/10.1145/nnnnnnnn>.
17. Ray PP, Pratim PR. A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions 2025. <https://doi.org/10.36227/techrxiv.174495492.22752319/v1>.
18. Anthropic. Introducing the Model Context Protocol 2024. <https://www.anthropic.com/news/model-context-protocol> (accessed June 17, 2025).
19. Mavroudis V. LangChain. HAL Open Science 2024. <https://doi.org/10.20944/preprints202411.0566.v1>.
20. CAMICON, MIDUVI. Norma ecuatoriana de la construcción - NEC: NEC-SE-MP - Mampostería estructural. Quito: 2014.
21. American Society of Civil Engineers. Minimum Design Loads and Associated Criteria for Buildings and Other Structures, ASCE/SEI 7-22. Reston: 2022.
22. Jiang G, Chen J. Efficient fine-tuning of large language models for automated building energy modeling in complex cases. *Autom Constr* 2025;175:106223. <https://doi.org/10.1016/j.AUTCON.2025.106223>.
23. Jurišević N, Kowalik R, Gordić D, Novaković A, Vukašinović V, Rakić N, et al. Large Language Models as Tools for Public Building Energy Management: An Assessment of Possibilities and Barriers. *International Journal for Quality Research* 2025;19. <https://doi.org/10.18421/IJQR19.03-09>.

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.