Leveraging AI in engineering knowledge work: a framework for evaluating effective use cases

Jasmine Badiee Konrad
NEI Electric Power Engineering
Lakewood, Colorado, USA
https://orcid.org/0000-0001-6679-7698

Jake Wiggins
NEI Electric Power Engineering
Lakewood, Colorado, USA
jwiggins@neieng.com

Abstract — **Power systems engineering firms are facing pressure to meet accelerated timelines without compromising technical quality. At the same time, the rapid emergence of large language models (LLMs) has prompted a surge of AI-based solutions aimed at relieving these operational demands. In engineering knowledge work, rapid adoption of such tools has occurred largely in the absence of structured frameworks for identifying appropriate use cases, resulting in inconsistent and sometimes ineffective deployment. This paper introduces a methodology for determining contexts in which AI tools can genuinely enhance engineering knowledge work, offering a practice-oriented solution tailored to the power systems engineering domain. The methodology serves as guidance for integrating AI tools in ways that complement engineering expertise to augment design processes, knowledge sharing, and technical workflows. The framework is expected to augment design reasoning, accelerate decision support, and standardize repetitive tasks.**

Keywords— human-AI collaboration, engineering knowledge work, task taxonomy, AI governance, engineering design process

I. Introduction

Electricity demand in the United States is increasing, driven in part by the expansion of data center infrastructure [1]. In 2023, data centers accounted for approximately 4.4% of total U.S. electricity consumption, with projections ranging from 6.7% to 12.0% by 2028, depending on growth scenarios [2]. While data center growth has been ongoing due to increasing digitization [3], recent advances in large-scale artificial intelligence (AI) models have further accelerated demand for high-density computing facilities [4] intensifying load growth.

This accelerating demand for energy infrastructure has put pressure on power systems firms that design and deliver it. The strain on the energy industry is visible in growing interconnection queues [5], [6], [7], [8], [9], [10], and a workforce that must expand rapidly to meet demand. These pressures compound one another: a thin engineering talent pool slows project execution, which in turn lengthens queue backlogs and defers the capacity additions that are urgently needed. IEEE-PES and Kearney estimate that the engineering workforce in the power systems industry may need to double by 2030 [11], a target also strained by a retiring senior cohort [12].

For these reasons, engineering firms must innovate to remain competitive [12], and many are exploring ways to accelerate delivery without sacrificing quality. Due to the efficiency gains achieved in other industries, large language models (LLMs) are a candidate for streamlining work in power systems engineering. Unlike conventional engineering software, which applies fixed rules to produce repeatable outputs for the same inputs, LLMs generate responses by predicting likely continuations from patterns learned from their training data [13], [14]. LLMs pose specific risks to the power systems industry due to a lack of domain-specific training data [15], [16], cybersecurity vulnerabilities [17], and the "black box" nature [16] of AI output. In engineering, these risks are amplified because liability for technical decisions remains with the engineer and the organization, not with the tool [18]. Therefore, a structured method for evaluating AI suitability at the task level is required to avoid ad hoc deployment under high-liability conditions.

Prior research has examined LLM use in adjacent engineering contexts, including code querying and adherence [19], building information modeling (BIM) [20] and cost estimation [21], [22]. More specifically, in a power systems setting, studies exist regarding load forecasting [23], [24], [25], fault detection [26], and grid optimization [27], [28], [29]. Related literature exists regarding AI and construction, and the broader AEC sector [19], [30], [31], [32], [33], [34], [35]. What remains underdeveloped, however, is a task-level framework for determining which tasks LLM tools are best suited for at engineering firms.

The existing literature addresses what tasks LLMs have performed in engineering; however, it does not address how an organization should decide whether to use them for a given task. This paper presents a practical framework for task-level AI deployment decisions within electric power engineering. The framework is designed to help engineering organizations distinguish where LLM-based tools can automate work, where they can productively accelerate analysis, and where they should remain limited to assistive roles under explicit human governance.

II. Framework

A. Task Classification Dimensions

To build a framework that complements engineering expertise, we introduce a classification of engineering tasks

along two dimensions: degree of problem structure and degree of outcome convergence. The degree of problem structure was introduced by Herbert Simon and operationalized by Gorry and Scott Morton. Structure measures the extent to which a task's inputs, constraints, solutions, and evaluation criteria are explicitly defined [36], [37]. We employ a continuum [38] from structured to unstructured to classify engineering tasks. Tasks with well-defined inputs, explicit constraints, established evaluation criteria, and a clear solution procedure will fall closer to the structured end of the spectrum [36], [37]. At the unstructured end, problems are more ambiguous, information may be incomplete, and outputs are dependent on practitioner judgment [36], [37]. Problem structure relates only to the problem statement and inputs, whereas outcome convergence measures the properties of a task's output(s).

The degree of outcome convergence measures the breadth of outputs given a single set of inputs. A high level of outcome convergence indicates that engineers will produce similar outputs given similar inputs for a task, whereas a low level suggests a multitude of defensible outputs. This distinction is especially useful in engineering, where multiple valid solutions to a problem can depend on the engineer's experience and expertise. As with problem structure, outcome convergence is best understood as a continuum [38] rather than a binary label.

B. Scoring Rubric

To reduce subjectivity in task placement, we propose a standardized scoring rubric. Each axis is evaluated using five diagnostic questions scored discretely on a scale of -2 to 2, with -2 being the least convergence and structure and 2 being high convergence and structure:

- +2: strongly aligned with structured or high convergence
- +1: moderately aligned with structured or high convergence
- 0: mixed, context-dependent, or unknown
- -1: moderately aligned with unstructured or low convergence
- -2: strongly aligned with unstructured or low convergence

1) X-Axis: The Degree of Problem Structure

Tasks are assessed along five dimensions, with each dimension receiving a score of +2 for a definitive yes response and −2 for a definitive no.

1. Input completeness: Are all required inputs available to start the task?
2. Problem clarity: Can the task be performed without requiring interpretation?
3. Procedural availability: Does a procedure exist to complete the task?
4. Evaluation criteria: Are there criteria to know if your output is correct?
5. Tacit knowledge: Can the task be completed successfully without undocumented knowledge?

Input completeness measures if the problem's inputs are readily available and unambiguous. If essential information must be uncovered before beginning the task, this dimension should receive a lower score. Problem clarity refers to the engineer's ability to understand the task's definition. Procedural availability is rated based on whether a clear procedure exists to perform the task. Evaluation criteria relate not to the outcome itself, but to whether the engineer has a clear way to determine whether the task output is correct. The tacit knowledge dimension measures the amount of tacit, undocumented/implicit knowledge that the engineer must possess to complete the task. Taken together, these dimensions reveal the level of problem structure for an engineering task.

2) Y-Axis: The Degree of Outcome Convergence

The Y-Axis questions are evaluated the same as the X-Axis, with a score of +2 for a definitive yes response and −2 for a definitive no.

1. Solution uniqueness: Will the task produce a singular correct answer, or are there multiple defensible solutions?
2. Independent reproducibility: Would qualified engineers working independently with the same inputs produce relatively equivalent outputs?
3. Assumption sensitivity: Will varied assumptions produce materially the same outcome?
4. Validation autonomy: Can the output be verified without expert interpretation?
5. Reasoning transparency: Can the chain of reasoning for solving the task be explicit?

Solution uniqueness spans the output landscape, with narrow landscapes scoring higher on the scale. Independent reproducibility is similar to solution uniqueness but specifically measures if multiple engineers arrive at the same answer given the same inputs. Assumption sensitivity is particularly important for engineering, where multiple assumptions can be valid and affect the output of a task. Validation autonomy is a verification check that reviews whether an output can be deemed acceptable without expert input. The reasoning transparency dimension measures the "black box" nature of an output and whether someone could review the output of a task and clearly draw conclusions about the reasoning that led to the output. Each dimension relates to a specific aspect of evaluating the outcome of a task, and, when taken together, provides an estimate of the expected level of convergence.

For each axis, the outputs of the 5 questions are added to get a value on a scale of -10 to +10, where -10 is highly unstructured or divergent, and +10 is highly structured or convergent. For all engineers surveyed, the X and Y values per task are averaged. Lastly, this average X-Y pair per task is plotted to create a task classification map, which indicates which AI deployment methods are best suited to the task's nature. This framework will be highly unique to the organizations that employ it, as scores will reflect the operational context of the organization. It is intended as a portable method; the specific values shown in our results are an initial instantiation representative of the authors' organization and experiences.

Table I outlines an illustrative example of the rubric applied to a protection coordination study, a common task in power systems engineering[a].

TABLE I. RUBRIC SCORING FOR PROTECTION COORDINATION STUDY

Axis	Dimension	Score	Rationale
Problem Structure	Input Completeness	+1	Core data generally available, potential for some estimation.
	Problem Clarity	+1	Study objective is clear once the one-line and protection philosophy are established. Philosophy itself may require iteration.
	Procedural Availability	+1	Time-current coordination methodology is well-established via IEEE C37 standards and standard software, though overall workflow is experience-guided.
	Evaluation Criteria	+1	Quantitative acceptance criteria exist (e.g., coordination time intervals, arc-flash limits), but margin tradeoffs require engineering judgment.
	Tacit Knowledge	−1	Effective coordination requires undocumented knowledge including site-specific preferences, relay model familiarity, and planned system expansions.
Outcome Convergence	Solution Uniqueness	−1	Multiple valid coordination schemes exist; engineers prioritizing speed vs. selectivity will select different curve shapes and time dials.
	Reproducibility	0	Two competent engineers will produce functionally equivalent schemes, but specific settings will differ, especially for complex multi-source systems.
	Assumption Sensitivity	−1	Conclusions are sensitive to assumed fault contributions, arc-flash assumptions, and which operating configurations are considered.
	Validation Autonomy	+1	Coordination intervals and arc-flash energy levels are quantitatively checkable, but overall scheme adequacy requires expert interpretation.
	Reasoning Transparency	+1	Decisions are documentable on time-current curves and settings spreadsheets, though higher-level tradeoff reasoning resists full externalization.

[a] Composite scores: Problem Structure (X) = +3; Outcome Convergence (Y) = 0

C. *Quadrant Interpretation*

Tasks are plotted in the structured–unstructured versus convergent–divergent space to determine quadrant placement. Parasuraman, Sheridan, and Wickens established that automation should not be treated as an all-or-nothing proposition but rather applied at different levels to different function types [39]. Extending that principle, we define four deployment modes[b], each representing a distinct level of AI involvement (Table II).

TABLE II. AI DEPLOYMENT MODE BY QUADRANT CLASSIFICATION

Quadrant	Deployment Mode	Description
Structured–Convergent (+X, +Y)	Automate	Full AI execution of a task, including verification. The engineer confirms the validity of the output and its verification.
Structured–Divergent (+X, −Y)	Accelerate	AI used to generate divergent output options from structured inputs, with all deterministic verification. The engineer selects the best option and confirms the output's validity.
Unstructured–Convergent (−X, +Y)	Formalize	AI helps structure or clarify the task; engineers use conventional methods to perform the task
Unstructured–Divergent (−X, −Y)	Explore	AI is leveraged as an analysis tool to explore various methods and their respective outputs. The engineer owns all assumptions and decisions.

[b] Deployment modes are determined by the location of task on the classification map based on respondent scoring.

Automation is applied to highly structured and convergent tasks due to the clarity afforded by a structured methodology and the narrow set of correct outputs. LLMs have been shown to produce indeterminate outputs [40], [41]. However, this can be ameliorated by creating a workflow, built by engineers, that includes deterministic measures, such as external data sources, code, and validation steps, so that task completion and verification can remain explicit and auditable. These workflows enable AI to invoke external capabilities rather than relying on probabilistic text-based methods to solve complex engineering problems [42]. Even in automation scenarios, the engineer should complete a final verification of the output.

Acceleration can be employed for tasks characterized as structured-divergent. These tasks have a generally straightforward methodology but may have a range of possible outputs depending on a variety of valid assumptions. For tasks in this category, AI can be utilized to expose a broader solution landscape, allowing engineers to review outcomes of their assumptions faster than using traditional methods. However, engineers should remain involved in reviewing the outputs and using their best judgment and expertise to complete the task. As should be expected, this is a higher level of verification than required with automation tasks.

While automate and accelerate tasks involve engineers as verifiers, the formalization and exploration modes augment an engineer's ability to perform the task itself. In formalization, AI can be leveraged to clarify an ill-defined task into something actionable [43]. Operationalization of formalization will vary by engineer and task, but a few examples may include working backwards from a known solution, performing a gap analysis for missing data, and brainstorming methodology options [44]. The engineer's domain judgment is essential for "scaffolding" the AI and evaluating its output [45] before proceeding to task

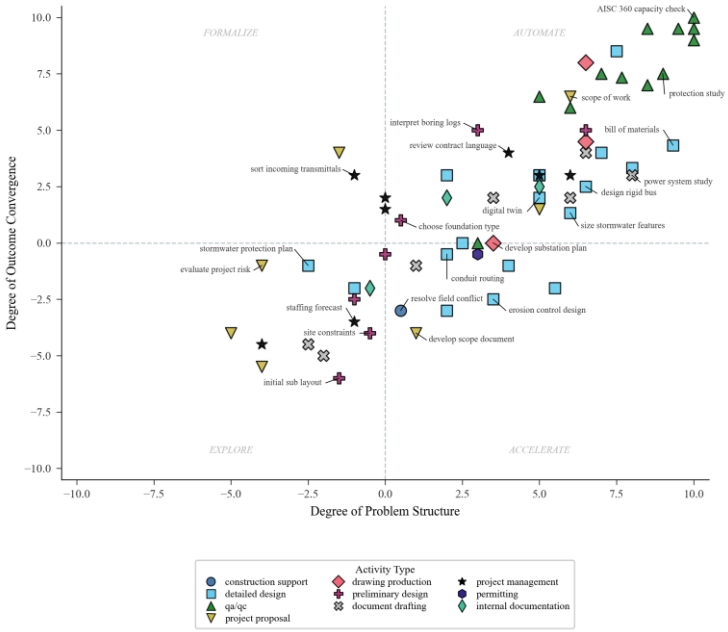

Fig. 1. Task classification map by activity. (*Distribution of 67 engineering tasks across the framework's four deployment quadrants plotted by average degree of problem structure (X) and average degree of outcome convergence (Y). Each bubble represents a single task, with the number indicating activity type and color intensity representing the consequence-of-error score (1 = low, 3 = high).*)

completion, since the ambiguity lies in the problem setup rather than the solution.

Lastly, exploration is the mode applied to unstructured-divergent tasks, where neither methodology nor output is well-defined. These tasks resist automation and acceleration because they require the engineer to exercise judgment across both methodology and output simultaneously. Rather than directing AI toward a known destination, engineers can use AI as a brainstorming partner [46]: surfacing possibilities, stress-testing assumptions, and iteratively narrowing the problem space. In this mode, the engineer's expertise and judgment are the primary drivers, with AI in a supporting role. As tasks become less structured and more divergent, the engineer's role shifts from verifying AI outputs to directing and ultimately leading the work, with AI serving an increasingly supporting function.

D. Consequences of Error

Engineers must manage and control risk when making decisions. Given the ubiquity of risk in the engineering discipline, we augment the two-dimensional task framework with a third component: the consequence of error. Consequence of error captures the potential impact if an AI-generated output is wrong and passes forward unchecked. While problem structure and outcome convergence describe the nature of a task, the consequences of error indicate the severity of performing that task incorrectly, thus indicating the level of required validation of an AI output.

The consequence of error is scored on a 3-point scale, with 1 indicating low risk and 3 indicating high risk. Risk distinguishes two tasks occupying the same quadrant by their oversight requirements: a task in the unstructured-divergent quadrant with low risk may be worth enhancing with AI, whereas a structured-convergent task with high risk may never be automated. As with the axis scores, consequence thresholds should be calibrated to organizational context. What constitutes high financial exposure for a small engineering firm with limited scopes will have vastly different tolerances than the same judgment made in utility, large infrastructure, or industrial settings.

III. Framework Application: Results and Discussion

As an initial validation of our framework, we applied it to a small convenience sample of three engineers at a single US-based power systems firm. The goal of this analysis was to apply the framework to a domain where ground-truth expectations were available, allowing us to assess whether classifications aligned with practitioner intuition. Fig. 1 shows the distribution of 67 tasks across the framework's four quadrants.

Quality assurance and quality control (QA/QC) activities are clustered at the top of the first quadrant (Fig. 1), suggesting they are promising candidates for automation. Examples of tasks in this category include checking steel components against known load capacities, determining the level of runoff from a stormwater event, and verifying electrical clearances of a substation. Interestingly, the most structured and convergent tasks were rated as having the highest level of risk (Fig. 2). This may suggest the need for a cautious posture toward full automation of these tasks despite strong technical feasibility. QA/QC has one outlier: the interdisciplinary review at the 60% milestone, which is on the border between quadrants 1 and 2.

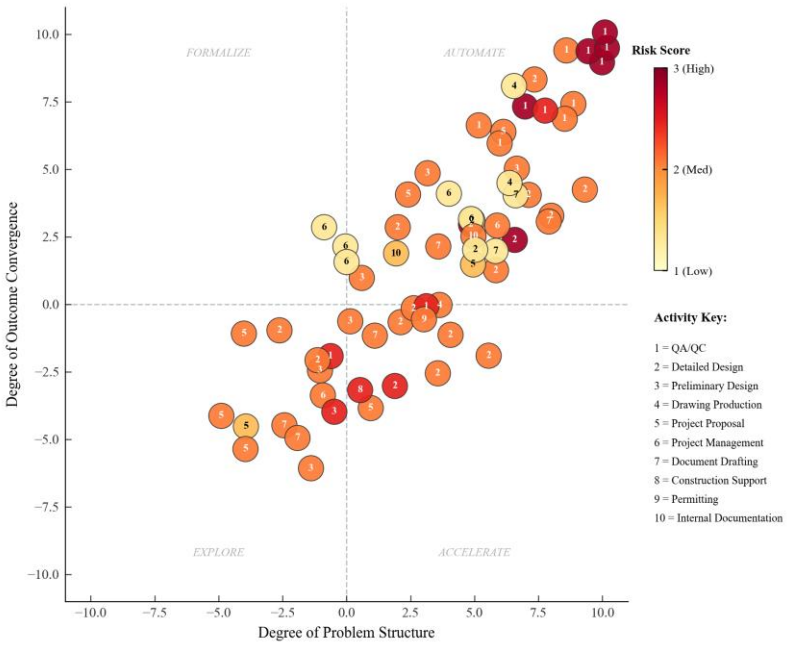

Fig. 2. Task classification by risk. (*Distribution of 67 engineering tasks across the framework's four deployment quadrants plotted by average degree of problem structure (X) and average degree of outcome convergence (Y). Each bubble represents a single task, with the number indicating activity type and color intensity representing the consequence-of-error score (1 = low, 3 = high).*)

Intuitively, this makes sense, as engineers have processes that inform their QA/QC methods, but they may yield various outcomes

Items in the *accelerate* quadrant (Q2) include responding to RFIs, providing construction support, and permitting (Fig. 1). These tasks are relatively structured but would yield varied outputs depending on the engineers and the assumptions made when completing them. Risk for tasks in this quadrant is lower than in the *automation* quadrant, hovering at between 2 and 2.5. This suggests that this power system engineering firm may benefit from using AI as an accelerator rather than automating routine tasks.

Three project management tasks are classified in or on the border of the *formalize* quadrant: class 1 cost estimates, class 3 cost estimates, and incoming transmittal sorting by action-path. This suggests that there is generally a correct way to perform this sorting, but that the knowledge related to that task is likely tacit. Relative to the other quadrants, *formalize* remains sparse, which may partly reflect the experience level of our sample. More experienced engineers may perceive problems as inherently structured due to accumulated implicit knowledge, effectively collapsing what might otherwise appear as unstructured tasks into the right half of the framework. Consistent with this interpretation, more experienced engineers in our sample had a higher median structure score (Supplemental Fig. 1).

As expected, detailed design tasks tend to cluster toward the structured end of the problem structure axis relative to preliminary design tasks, which skew toward the unstructured end. This is intuitive: preliminary design requires resolving a greater number of ambiguous decisions with limited information, while detailed design iterates on an already established design direction, constrains the solution space, and makes the problem inherently more structured.

Tasks with both the highest and lowest risk scores coexist within the automate quadrant (Fig. 2). For example, QA/QC tasks such as verifying electrical clearances carry high scores, while internal documentation tasks occupy the lower end of the risk spectrum despite similar quadrant placement. This suggests that problem structure and outcome convergence alone do not determine consequentiality; the stakes of an incorrect output vary substantially even among tasks that share similar structural characteristics.

Drawing production is the only activity type in which no tasks produce an unstructured classification (Supplemental Fig. 2). This is consistent with the nature of the work: engineers rely on CAD software with embedded workflows and standardized functions, which constrain the process and produce predictable, verifiable outputs. Drawing production may therefore represent a category in which structured AI assistance is both technically feasible and relatively low-friction to implement.

Consequences of errors do not scale cleanly with either problem structure or outcome convergence. In practice, it depends more on what downstream decisions the task influences. A breaker interrupting-duty calculation, for example, is highly structured and highly convergent, yet an error can propagate directly into equipment specification and fault-clearing risk. By contrast, an internal brainstorming memo may be low-consequence even if it is unstructured and low-convergence. The overlay, therefore, matters because it prevents organizations from treating technically checkable work as automatically low risk.

Risk adds a necessary scale that clarifies why AI suitability is not uniform across engineering tasks. Even if an AI system can generate plausible outputs across all regions, the cost and risk of verifying those outputs can vary significantly. In high-risk scenarios, verification effort may approach or exceed the effort required to perform the task manually, erasing the productivity advantage that initially motivated AI adoption. To view these results in an interactive setting, visit taskplot.neieng.com.

IV. Limitations

Several limitations should be considered when interpreting the results of this study. First, the analysis sample consists of three engineers at a single US-based power systems firm. While this sample was sufficient for an operationalization of the framework, the classifications produced reflect the unique operational context, tacit knowledge, and norms of one organization. We expect that results will differ across engineering firms, disciplines, and experiences. Generalization should be made cautiously until broader validation is completed.

Second, all respondents in this study share the same disciplinary background. Power systems engineering encompasses a diverse set of tasks, and engineers from different specializations may produce meaningfully different classifications of overlapping tasks. Furthermore, our analysis was limited in the range of expertise. In particular, the finding that the *formalize* quadrant remains sparse, for example, may partly reflect an expertise bias rather than a property of power systems engineering work more broadly. Familiarity with a task may inflate perceived structure, as an engineer with deep tacit knowledge may experience a task as more defined than it would appear to a less experienced colleague (Supplemental Fig. 2).

Finally, the framework produces a static classification. AI capability is evolving rapidly, and a task that is unstructured or divergent today may become more tractable as the technology matures, training data improves, or organizational workflows are redesigned around AI assistance. Classifications should therefore be treated as time-sensitive assessments rather than fixed designations, and organizations are encouraged to reassess task placements periodically as both their workflows and available tools evolve.

V. Conclusions and Future Work

Using AI in engineering work is no longer hypothetical but rather an operational reality. Organizations have begun navigating this integration without adequate frameworks to guide deployment decisions, resulting in inconsistent and sometimes ineffective adoption [47]. This paper addresses that gap by introducing a practice-oriented methodology for evaluating where LLM-based tools are suitable across engineering knowledge work, with particular attention to the power systems engineering domain.

The framework classifies engineering tasks along two dimensions: the degree of problem structure and the degree of outcome convergence. Together, these components provide a repeatable mechanism for positioning tasks within a decision space that informs the best AI application method. We suggest that full AI automation should not be applied broadly to engineering tasks due to their varying natures. Our framework identifies where AI can automate work reliably, where it can productively accelerate analysis, where it can formalize requirements, and where it can be used to augment human analysis. In all of these scenarios, including full automation, the engineer remains integral.

An initial application of the framework to a convenience sample of three engineers at a single power systems firm produced classifications that aligned with practitioner intuition across multiple activity types. QA/QC tasks clustered in the automate quadrant yet carried the highest risk scores in the sample. This illustrates our framework's core value: technical feasibility and deployment appropriateness are not equivalent. Detailed design tasks scored as more structured than preliminary design tasks, consistent with domain expectations. The accelerate quadrant's lower-risk profile relative to the automate quadrant suggests that AI augmentation, rather than full automation, may be a more prudent near-term deployment posture for many engineering firms. Drawing production emerged as the most uniformly structured activity type, making it a natural candidate for structured AI assistance.

Empirical validation remains the most immediate and important direction for future work. A structured study in which practitioners from multiple organizations independently score the same task set would establish whether the framework yields stable classifications across raters and organizational contexts and reveal where the rubric requires additional calibration. A second line of future work should evaluate post-deployment outcomes: whether organizations that apply the framework make better deployment decisions, realize durable productivity gains, and avoid inappropriate use. Longitudinal application of the framework would also allow organizations to track how classifications shift as AI capabilities mature and internal workflows evolve, converting the framework from a one-time diagnostic into an ongoing governance instrument.

Acknowledgment

The authors thank the chief engineers at NEI Electric Power Engineering, Andrew Merritt, Carson Bates, and Mark Lanphier, for their assistance in guiding the brainstorming process and for reviewing the framework.

References

[1] U.S. Energy Information Administration, “Short-Term Energy Outlook,” Washington, DC, Feb. 2026.

[2] A. Shehabi, A. Newkirk, and S. J. Smith, “2024 United States Data Center Energy Usage Report,” Berkeley, Dec. 2024.

[3] A. Shehabi, S. J. Smith, E. Masanet, and J. Koomey, “Data center growth in the United States: Decoupling the demand for services from electricity use,” Environmental Research Letters, vol. 13, no. 12, Dec. 2018, doi: 10.1088/1748-9326/aaec9c.

[4] B. Lee et al., “Generational Growth AI, data centers and the coming US power demand surge,” 2024.

[5] J. Rand et al., “Queued Up: 2025 Edition – Characteristics of Power Plants Seeking Transmission Interconnection As of the End of 2024,” Dec. 2025. doi: 10.2172/3008763.

[6] W. Gorman et al., “Grid connection barriers to renewable energy deployment in the United States,” Joule, vol. 9, no. 2, Feb. 2025, doi: 10.1016/j.joule.2024.11.008.

[7] G. Brown, B. Chan, R. Clune, and Z. Cutler, “Upgrade the grid: Speed is of the essence in the energy transition,” McKinsey & Company Insights.

[8] H. Bawa, A. Delgado, and N. Aoun, “How to accelerate grid infrastructure deployment for an electrified future,” World Economic Forum.

[9] T. Y. Elete, E. Onyinye Nwulu, O. V. Erhueh, O. A. Akano, and A. T. Aderamo, “Impact of Front End and Detailed Design Engineering on Project Delivery Timelines and Operational Efficiency in the Energy Sector,” 2024. [Online]. Available: www.ijerd.com

[10] T. L. Keefe, K. Hardin, and J. Nagdeo, “2026 Power and Utilities Industry Outlook,” Deloitte Center for Energy & Industrials.

[11] “The future of the energy workforce,” 2024.

[12] M. Crawford, “6 Top Challenges Facing Engineering Firms in 2023,” The American Society of Mechanical Engineers.

[13] T. B. Brown et al., “Language Models are Few-Shot Learners,” Jul. 2020.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training.” [Online]. Available: https://gluebenchmark.com/leaderboard

[15] Q. Yao, F. Fang, Y. Chen, J. Liu, H. Mo, and Y. Ao, “AI Large Models for Power System: A Survey and Outlook,” IET Smart Energy Systems, vol. 1, no. 1, pp. 3–21, Jun. 2025, doi: 10.1049/ses2.70000.

[16] S. Majumder et al., “Exploring the Capabilities and Limitations of Large Language Models in the Electric Energy Sector,” Jun. 2024, doi: 10.1016/j.joule.2024.05.009.

[17] J. Ruan et al., “Applying Large Language Models to Power Systems: Potential Security Threats,” Jan. 2024.

[18] D. L. Van Bossuyt, A. Dong, I. Y. Tumer, and L. Carvalho, “On measuring engineering risk attitudes,” Journal of Mechanical Design, vol. 135, no. 12, 2013, doi: 10.1115/1.4025118.

[19] F. Yang and J. Zhang, “Prompt-based automation of building code information transformation for compliance checking,” Autom. Constr., vol. 168, p. 105817, Dec. 2024, doi: 10.1016/j.autcon.2024.105817.

[20] P. T. Koh, H. Xue, J. Ma, and J. C. P. Cheng, “Cost-effective and minimal-intervention BIM information retrieval via condensed multi-LLM agent code generation,” Autom. Constr., vol. 181, p. 106585, Jan. 2026, doi: 10.1016/j.autcon.2025.106585.

[21] A. Ghasemi and F. Dai, “Can ChatGPT assist in cost analysis and bid pricing in construction estimating? A pilot study using a bridge rehabilitation project,” Smart Construction, 2024, doi: 10.55092/sc20240009.

[22] P. Parsafard, O. Elezaj, D. Ekundayo, E. Vakaj, M. Parmar, and M. Ahmad Wani, “Automation in Construction Cost Budgeting using Generative Artificial Intelligence,” in Proceedings of the International Conference on Industrial Engineering and Operations Management, Michigan, USA: IEOM Society International, Feb. 2024. doi: 10.46254/AN14.20240466.

[23] M. Jin et al., “Time-LLM: Time Series Forecasting by Reprogramming Large Language Models,” Jan. 2024.

[24] W. Liao, Z. Yang, M. Jia, C. Rehtanz, J. Fang, and F. Porté-Agel, “Zero-Shot Load Forecasting with Large Language Models,” Nov. 2024.

[25] M. Gao, S. Zhou, W. Gu, Z. Wu, H. Liu, and A. Zhou, “A General Framework for Load Forecasting based on Pre-trained Large Language Model,” Sep. 2024.

[26] L. Jing and A. Rahman, “Fault Diagnosis in Power Grids with Large Language Model,” Jul. 2024.

[27] Y. Cheng et al., “A large language model for advanced power dispatch,” Sci. Rep., vol. 15, no. 1, p. 8925, Mar. 2025, doi: 10.1038/s41598-025-91940-x.

[28] F. Bernier, J. Cao, M. Cordy, and S. Ghamizi, “PowerGraph-LLM: Novel Power Grid Graph Embedding and Optimization With Large Language Models,” IEEE Transactions on Power Systems, vol. 40, no. 6, pp. 5483–5486, Nov. 2025, doi: 10.1109/TPWRS.2025.3596774.

[29] Z. Yan and Y. Xu, “Real-Time Optimal Power Flow With Linguistic Stipulations: Integrating GPT-Agent and Deep Reinforcement Learning,” IEEE Transactions on Power Systems, vol. 39, no. 2, pp. 4747–4750, Mar. 2024, doi: 10.1109/TPWRS.2023.3338961.

[30] L. Ma et al., “Adopting Large Language Models in the Construction Industry: Drivers, Barriers, and Strategic Implications from China,” Buildings, vol. 15, no. 23, Dec. 2025, doi: 10.3390/buildings15234296.

[31] S. O. Abioye et al., “Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges,” Dec. 01, 2021, Elsevier Ltd. doi: 10.1016/j.jobe.2021.103299.

[32] S. J. Badhan and R. Samsami, “Artificial Intelligence (AI) in Construction Safety: A Systematic Literature Review,” Nov. 01, 2025, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/buildings15224084.

[33] H. Martin, J. James, and A. Chadee, “Exploring Large Language Model AI tools in Construction Project Risk Assessment: Chat GPT Limitations in Risk Identification, Mitigation Strategies, and User Experience,” J. Constr. Eng. Manag., vol. 151, no. 9, Sep. 2025, doi: 10.1061/jcemd4.coeng-16658.

[34] L. Yang, G. Allen, Z. Zhang, and Y. Zhao, “Achieving On-Site Trustworthy AI Implementation in the Construction Industry: A Framework Across the AI Lifecycle,” Buildings, vol. 15, no. 1, Jan. 2025, doi: 10.3390/buildings15010021.

[35] P. Ghimire, K. Kim, and M. Acharya, “Opportunities and Challenges of Generative AI in Construction Industry: Focusing on Adoption of Text-Based Models,” Buildings, vol. 14, no. 1, Jan. 2024, doi: 10.3390/buildings14010220.

[36] H. A. Simon, The new science of management decision. New York: Harper & Brothers, 1960. doi: 10.1037/13978-000.

[37] G. A. Gorry and M. S. S. Morton, “A FRAMEWORK FOR MANAGEMENT INFORMATION SYSTEMS* by,” 1971.

[38] H. Mintzberg, D. Raisinghani, and A. Theoret, “The Structure of ‘Unstructured’ Decision Processes,” Adm. Sci. Q., vol. 21, no. 2, p. 246, Jun. 1976, doi: 10.2307/2392045.

[39] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “Situation Awareness, Mental Workload, and Trust in Automation: Viable, Empirically Supported Cognitive Engineering Constructs,” J. Cogn. Eng. Decis. Mak., vol. 2, no. 2, pp. 140–160, Jun. 2008, doi: 10.1518/155534308X284417.

[40] S. Ouyang, J. M. Zhang, M. Harman, and M. Wang, “An Empirical Study of the Non-Determinism of ChatGPT in Code Generation,” ACM Transactions on Software Engineering and Methodology, vol. 34, no. 2, Jan. 2025, doi: 10.1145/3697010.

[41] M. Lee, P. Liang, and Q. Yang, “CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities,” in CHI Conference on Human Factors in Computing Systems, New York, NY, USA: ACM, Apr. 2022, pp. 1–19. doi: 10.1145/3491102.3502030.

[42] OpenAI, “Function Calling,” OpenAI Developers. Accessed: Mar. 08, 2026. [Online]. Available: https://developers.openai.com/api/docs/guides/function-calling

[43] J. I. Saadi and M. C. Yang, “Generative Design: Reframing the Role of the Designer in Early-Stage Design Process,” Journal of Mechanical Design, vol. 145, no. 4, Apr. 2023, doi: 10.1115/1.4056799.

[44] E. M. Botero and J. T. Smart, “deepSPACE: Generative AI for Configuration Design Space Exploration,” 2024.

[45] A. Teixeira de Melo et al., “An AI tool for scaffolding complex thinking: challenges and solutions in developing an LLM prompt protocol suite,” Cognition, Technology & Work, vol. 27, no. 3, pp. 651–693, Sep. 2025, doi: 10.1007/s10111-025-00817-6.

[46] S. Wadinambiarachchi, R. M. Kelly, S. Pareek, Q. Zhou, and E. Velloso, “The Effects of Generative AI on Design Fixation and Divergent Thinking,” in Conference on Human Factors in Computing Systems - Proceedings, Association for Computing Machinery, May 2024. doi: 10.1145/3613904.3642919.

[47] S. Heo and S. Na, “Ready for departure: Factors to adopt large language model (LLM)-based artificial intelligence (AI) technology in the architecture, engineering and construction (AEC) industry,” Results in Engineering, vol. 25, p. 104325, Mar. 2025, doi: 10.1016/j.rineng.2025.104325.