Transient Stability Assessment via LLM-Driven Simulation and Neural Architecture Search

Lianzhe Hu[1], Yu Wang[1], and Bikash Pal[2]
[1]School of Electrical Engineering, Chongqing University, Chongqing 400044, China
[2]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK
Email: hulianzhe@163.com, yu_wang@cqu.edu.cn, b.pal@imperial.ac.uk

***Abstract*—This paper presents an LLM-driven agentic workflow that streamlines transient stability assessment (TSA) by reducing manual effort in scenario specification, simulation execution, and data-driven model design. The workflow couples (i) an LLM-based simulation controller that translates natural-language disturbance descriptions into executable ANDES cases using prompt templates, retrieval-augmented generation (RAG), and error-feedback retries, and (ii) an LLM-guided neural architecture design module that iteratively proposes and evaluates candidate models under accuracy and real-time constraints. On the IEEE 39-bus system, the LLM-designed classifier achieves 93.71% test accuracy for four-class TSA with 4.78M parameters and <0.95 ms inference latency per sample, outperforming representative manually designed baselines. We further report scenario validity/coverage statistics and ablation results showing the benefits of retrieval, structured prompting, and feedback. The proposed workflow is positioned as a practical way to standardize and accelerate routine TSA studies after an initial expert setup of templates, knowledge sources, and search constraints.**

***Index Terms*—Large Language Models, Agentic Workflow, Power Systems, Transient Stability Assessment, Neural Architecture Search, Automated Scientific Discovery**

I. Introduction

Modern power grids are undergoing a fundamental transformation driven by global energy transition, characterized by increasing integration of high-penetration renewable energy sources (RES) and power-electronic-interfaced devices [1]. This shift introduces unprecedented complexities in network topology, dynamic characteristics, and stability mechanisms. Consequently, this transformation, while critical for a sustainable energy future, poses severe challenges to conventional transient stability assessment (TSA) methodologies, which were designed for traditional, centralized generation systems. The urgency of this issue is underscored by the frequent occurrence of related power system security incidents. Notable examples include the 2016 South Australia blackout, attributed to high wind power integration [2], and the 2025 Iberian Peninsula blackout [3], where an initial small-scale generation tripping event—against a backdrop of high renewable energy penetration—rapidly escalated into a large-scale cascading failure.

The transient stability of a power system refers to its ability to maintain synchronous operation after being subjected to significant disturbances [4]. Traditional power system stability assessment generally relies on mathematical and physical methods, such as time-domain simulation, energy function method [5], and analytical calculation. Although these methods can provide interpretable results, in modern power systems, they have very prominent limitations. The computational complexity of them will grow rapidly as the system scale increases and renewable energy is added. Scene generation requires a lot of expert knowledge and human intervention.

Meanwhile, with the rapid development of artificial intelligence theory and computing hardware, the technical route of conducting stability analysis based on artificial intelligence technology has gradually become obvious [6]. The model architecture has become increasingly complex, developing from shallow machine learning to complex deep neural networks [7]. Large language models are a very crucial branch in artificial intelligence. Its realization relies on algorithmic innovation in natural language processing, advancements in computing power, and the diffusion of big data [8]. These technological breakthroughs have significantly enhanced the ability of artificial intelligence systems to absorb knowledge, decompose complex tasks, and handle multimodal data [9]. By integrating LLM and professional knowledge in the field of power systems within a modular framework, these models can independently and accurately complete a wide variety of simulation tasks [10]. Technologies such as retrieval improvement generation, prompt engineering, and thought chains can integrate power system knowledge into LLM programs [11].

Although the application of large language models in the field of power systems has developed quite rapidly, up to now, their application in the stability assessment of power systems is still rather scattered and the application scope is also relatively limited. The existing methods generally only handle individual parts of the assessment workflow, such as data preprocessing, feature extraction or result interpretation, etc. Although neural architecture search has brought significant changes to the design of automated models in fields such as computer vision and natural language processing [12], its application in the stability assessment of power systems is still relatively limited. Currently, the deep learning methods used for transient stability assessment rely on artificially constructed architectures. This limits their adaptability to various power system configurations and operating conditions.

Prior studies have demonstrated LLM-assisted power-system simulation and feedback-driven multi-agent tool use. In contrast, our focus is a TSA-oriented end-to-end workflow that connects disturbance scenario generation, automated sim-

979-8-3315-8138-1

ulation/labeling, and architecture design for fast TSA inference under resource constraints. Accordingly, the contribution is at the workflow and orchestration level: we specify domain-constrained prompting and retrieval assets for transient simulation, and we empirically evaluate how an LLM-guided architecture search can improve accuracy-latency trade-offs on TSA data.

Building upon the aforementioned concepts and exploring the feasibility of LLM-driven applications in transient stability assessment (TSA), this paper makes three primary contributions by integrating LLMs as intelligent agents into a complete power system TSA workflow:

TSA-Oriented Agentic Workflow Orchestration: We present an end-to-end, LLM-driven workflow that connects disturbance specification, automated simulation and labeling, and fast TSA inference model development, aiming to reduce repetitive manual scripting and tuning in TSA studies.

Domain-Constrained LLM Simulation Control: We develop a simulation controller that leverages structured prompting, retrieval-augmented generation (RAG), and error-feedback retries to translate natural-language disturbance descriptions into executable ANDES cases under domain constraints (fault types, locations, timing, and parameter bounds).

LLM-Guided Architecture Design under Resource Constraints: We introduce an LLM-guided neural architecture design module that searches for architectures optimized for the TSA feature structure while jointly considering accuracy, model size, and inference latency, and we validate the resulting accuracy-latency improvements with ablations.

II. Methodology

A. LLM-Driven Simulation Controller

Our framework uses LLMs as intelligent simulation controllers to achieve automated scenario generation and data collection. The framework integrates three key components:

(1) Prompt Engineering: Multi-stage prompts guide the LLM to generate executable ANDES simulation scripts. Following chain-of-thought reasoning, we design sequential prompts: (a) role definition specifying the LLM as a power system expert, (b) architecture description detailing ANDES API and simulation parameters, and (c) feedback prompts for error correction. These prompts are infused with domain-specific knowledge to improve effectiveness.

(2) Customized RAG: Retrieves domain-specific knowledge including ANDES documentation, power system theory, and code examples. The RAG module breaks down natural language requests into sub-requests and maps them to simulation functions and parameters, ensuring greater factual consistency and reducing hallucinations.

(3) Feedback Loop: Iteratively corrects simulation errors through diagnostic prompts. When simulation fails, error information is fed back to the LLM for self-correction and retry. This feedback mechanism enhances the success rate of complex simulation tasks.

Stability assessment uses three mathematical criteria. The rotor angle stability criterion is:

$$\max_{i,j\in G, t\in[0,T]} |\delta_i(t) - \delta_j(t)| < 180^\circ \tag{1}$$

The voltage stability criterion is:

$$V_{min} \le V_i(t) \le V_{max}, \quad \forall i \in N, t \in [0,T] \tag{2}$$

The frequency stability criterion is:

$$|f(t) - f_0| \le \Delta f_{max}, \quad \forall t \in [0,T] \tag{3}$$

where $\delta_i(t)$ is the rotor angle of generator i, $V_i(t)$ is the voltage magnitude at node i, and $f(t)$ is the system frequency. The framework generates balanced datasets by systematically varying fault parameters, locations, and clearing times.

B. LLM-Driven Neural Network Designer

Traditional neural network design for power system stability assessment heavily relies on manual architecture engineering and hyperparameter tuning. We propose an LLM-driven NAS framework that leverages LLMs' semantic reasoning and code generation capabilities.

1) Search Space Formulation: The neural architecture search problem is formally defined as finding an optimal architecture $\alpha^* \in \mathcal{A}$ that maximizes a multi-objective function:

$$\alpha^* = \arg\max_{\alpha\in\mathcal{A}} \mathcal{F}(\alpha) = \arg\max_{\alpha\in\mathcal{A}} \{P(\alpha), -C(\alpha), -L(\alpha)\} \tag{4}$$

where $\mathcal{A}$ denotes the search space of all valid architectures, $P(\alpha)$ represents prediction accuracy, $C(\alpha)$ denotes computational cost (parameters, FLOPs), and $L(\alpha)$ represents inference latency. The search space is defined by:

$$\mathcal{A} = \{\alpha = (d_1, d_2, \ldots, d_k) \mid d_i \in \mathcal{D}_i, i = 1, \ldots, k\} \tag{5}$$

where each d_i represents a design decision (layer type, kernel size, activation function) and $\mathcal{D}_i$ is the corresponding choice set.

2) Collaborative LLM Framework: The LLM-NND framework employs three specialized LLM agents:

Navigator LLM: A stateful component with persistent memory across iterations. It maintains a history of evaluated architectures and their performance metrics $\mathcal{H} = \{(\alpha_i, P_i, C_i, L_i)\}_{i=1}^{t}$. The Navigator formulates and refines search strategies $\mathcal{S}_t$ based on accumulated feedback:

$$\mathcal{S}_t = \text{NavigatorLLM}(\mathcal{H}_{t-1}, P_{target}, \Lambda) \tag{6}$$

where P_{target} is the target accuracy and Λ represents resource constraints.

Generator LLM: A stateless component that translates the current strategy into concrete candidate architectures. Given a strategy $\mathcal{S}_t$, the Generator produces a set of candidate architectures:

$$\mathcal{C}_t = \text{GeneratorLLM}(\mathcal{S}_t) \tag{7}$$

979-8-3315-8138-1

The Generator focuses exclusively on the current strategy without retaining memory from previous iterations, enabling diverse exploration within the guided search space.

Coordinator Module: Manages the overall search process by verifying architecture legality, evaluating performance, and maintaining an archive of visited architectures $\mathcal{V}$. For each candidate architecture $\alpha_i \in \mathcal{C}_t$, the Coordinator computes:

$$\text{isValid}(\alpha_i) = \begin{cases} 1 & \text{if } \alpha_i \in \mathcal{A} \text{ and } \alpha_i \notin \mathcal{V} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

3) Iterative Search Algorithm: The LLM-based NAS process follows an iterative refinement strategy. At each iteration t:

$$\begin{aligned} \mathcal{C}_t &\leftarrow \text{GeneratorLLM}(\mathcal{S}_{t-1}) \\ \mathcal{R}_t &\leftarrow \{\text{Evaluate}(\alpha_i) \mid \alpha_i \in \mathcal{C}_t \setminus \mathcal{V}, \text{isValid}(\alpha_i)\} \\ \alpha^*, p^* &\leftarrow \arg \max_{(\alpha_i, p_i) \in \mathcal{R}_t} p_i \text{ s.t. } C(\alpha_i) \leq \Lambda \\ \mathcal{S}_t &\leftarrow \text{NavigatorLLM}(\mathcal{H}_{t-1} \cup \{(\mathcal{S}_{t-1}, \mathcal{R}_t)\}) \end{aligned} \tag{9}$$

The search terminates when either the target accuracy is achieved or the iteration limit is reached.

4) Advantages over Traditional NAS: Compared with traditional methods using evolutionary algorithms or reinforcement learning, LLM-based NAS has several theoretical advantages. Large language models can reason about architecture design principles at the semantic level and combine domain knowledge such as power system characteristics and computational constraints. LLMs naturally handle multiple competing objectives without explicit Pareto boundary calculations, significantly accelerating the search process.

5) LLM-NND Framework Algorithm: The LLM-based NAS process is formalized in Algorithm 1. Given data characteristics D, performance requirements R, and a maximum iteration count T, the framework intelligently searches for a neural network architecture and iteratively applies validation until the user's requirements are met.

Algorithm 1: LLM-NND Framework Algorithm

Input: Data features D, performance requirements R, maximum iterations T
Output: Optimal neural network architecture A^*
Initialize: $A^* \leftarrow \emptyset$, $P^* \leftarrow 0$;
for $t = 1$ **to** T **do**
  $A_t \leftarrow$ LLM generates architecture$(D, R, feedback_{t-1})$;
  $P_t \leftarrow$ Evaluate architecture(A_t, D);
  if $P_t > P^*$ **then**
    $A^* \leftarrow A_t$;
    $P^* \leftarrow P_t$;
  $feedback_t \leftarrow$ LLM generates feedback(A_t, P_t, R);
  if *convergence condition satisfied* **then**
    break;
return A^*;

III. Experimental Validation

A. Experimental Setup and Dataset

1) Dataset Construction and Characteristics: Using the LLM-Simulation pipeline on the IEEE 39-bus system, we generate a comprehensive multi-class dataset with **9,139** samples spanning four stability levels: Stable (51.21%), Voltage unstable (23.19%), Frequency unstable (12.80%), and Angle unstable (12.80%). The dataset captures voltage dynamics (39 buses over 101 time steps), generator dynamics (rotor angles and speeds of 10 synchronous machines), and transmission dynamics (power flows on 34 lines).

Fault scenarios include three-phase short circuits, single-line-to-ground faults, line outages and generator trips. To improve physical plausibility, fault locations cover all buses/lines, clearing times (50–500ms) and fault impedances are sampled within engineering-reasonable ranges, and each case must pass pre-fault power-flow initialization and post-fault sanity checks. Non-convergent cases are filtered via the error-feedback loop.

Table I summarizes the dataset statistics, showing the comprehensive coverage of stability classes and the high scenario validity achieved by the LLM-Simulation pipeline.

TABLE I: Dataset statistics (IEEE 39-bus, LLM-Simulation generated)

Parameter	**Value**	**Description**
System scale	39 buses, 10 gen.	Network complexity
Total samples	9,139	Multi-contingency scenarios
Time window	101 steps	5-second post-fault
Stable	4,680 (51.21%)	Normal operation
Unstable	4,459 (48.79%)	3 instability modes
Original features	808	Complete state variables
Selected features	300	Post feature selection
Scenario validity	92%	Valid convergence

The dataset comprises **9,139** balanced samples with 92% scenario validity and 94.8% parameter-setting accuracy, generated in approximately 15 minutes.

B. LLM-NND Design and Optimization

1) Multi-class Classification Results: On the 4-class stability assessment task (IEEE 39-bus system), the LLM-NND framework achieves a test accuracy of **93.71%** with the optimized traditional neural network architecture. This result represents a significant advancement over baseline approaches and demonstrates the effectiveness of LLM-guided architecture design. As shown in Fig. 1, the training and validation accuracy curves converge smoothly without severe overfitting, indicating stable optimization of the discovered architecture.

2) Comparative Analysis with Baseline Methods: **Baseline Method Comparison:** We evaluate the framework against several baseline methods using the same IEEE 39-bus dataset:

Deep Learning Architecture Comparison: For a fair comparison, all baseline models were re-implemented and

979-8-3315-8138-1

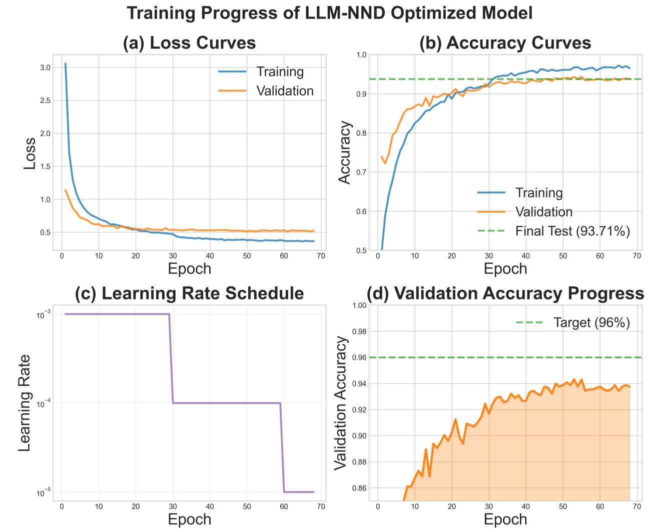

Fig. 1: Training and validation accuracy curves for multi-class stability assessment.

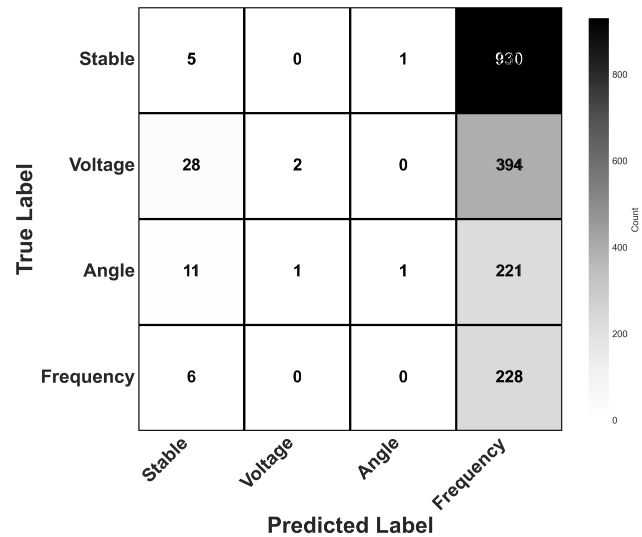

Fig. 2: Confusion matrix for 4-class stability assessment.

TABLE II: Performance comparison with baseline methods

Method	Acc.	Prec.	Recall	F1
GBDT	78.0%	68.8%	62.0%	63.6%
Random Forest	77.2%	68.0%	60.2%	61.6%
Simple MLP	77.0%	67.5%	59.0%	60.3%
LLM-NND	**93.71%**	**93.5%**	**93.71%**	**93.71%**

tuned on the same training/validation split under a common training budget with hyperparameter search over depth/width, learning rate, and dropout.

TABLE III: Architecture comparison (all baselines tuned)

Architecture	Acc.	F1	Params	Time
Basic CNN	76.5%	75.1%	850K	0.95ms
Basic LSTM	74.2%	72.8%	1.2M	1.2ms
Adv. LSTM	78.5%	77.2%	2.1M	1.5ms
Transformer	79.2%	78.1%	3.2M	2.1ms
DenseNet	80.05%	79.3%	25.9M	3.2ms
LLM-NND	**93.71%**	**93.71%**	**4.78M**	**0.95ms**

The LLM-NND framework achieves 13.66% improvement over DenseNet with 5.4× fewer parameters and 3.4× faster inference, demonstrating superior accuracy and computational efficiency.

As illustrated in Fig. 2, diagonal elements indicate correct classifications. Most misclassifications occur between adjacent stability classes, which is physically meaningful as these conditions share similar short-term dynamic signatures.

To further demonstrate the reproducibility and stability of our approach, we conduct multiple independent trials with different random seeds. Fig. 3 shows the validation accuracy curves across multiple trials, confirming that the LLM-NND framework consistently achieves high accuracy with minimal variance, indicating robust and reproducible performance.

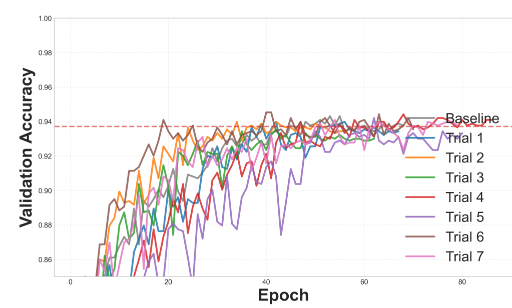

Fig. 3: Validation accuracy across multiple trials, confirming stable and reproducible results.

3) Renewable Energy Integration: We further validate the framework with wind power integration on the IEEE 39-bus system. The LLM-driven pipeline automatically configures wind generator models (REGCV2 virtual synchronous machine control), determines optimal injection buses based on system topology, and generates diverse fault scenarios under high renewable penetration.

Table IV presents the performance metrics across different renewable penetration levels, demonstrating the framework's capability to handle varying degrees of renewable integration while maintaining acceptable convergence rates.

Note that Conv. Rate refers to successful simulator initialization/DAE solution, not classification accuracy. At 50% penetration, converter-dominated dynamics increase numerical stiffness. The framework handles increased state-space dimensionality (124,308 features per sample) and complex interactions between conventional generators and renewable sources. Tighter parameter bounds and improved renewable

979-8-3315-8138-1

TABLE IV: Renewable energy integration performance

Penetration	Conv. Rate	Samples	Speed
10%	95.0%	285	0.365
30%	92.7%	278	0.463
50%	70.7%	212	0.374

templates can mitigate convergence issues; nevertheless, robust generation for very high-penetration cases remains a limitation.

C. Extended System Validation

To demonstrate scalability, we validate the framework on the IEEE 118-bus system (118 buses, 54 generators). The LLM-Simulation pipeline generates 3,970 balanced binary classification samples with 94% scenario validity. The LLM-NND framework achieves **96.66%** test accuracy on this larger system, automatically designing a hybrid CNN-dense architecture with 350K parameters. This result validates that the framework generalizes effectively across different system scales without manual architecture tuning.

IV. Conclusion

This work develops an LLM-driven agentic workflow that reduces manual scripting and iterative tuning in TSA studies by coupling LLM-controlled scenario generation with LLM-guided architecture design. On the IEEE 39-bus system, the discovered model achieves 93.71% four-class accuracy with $<$0.95 ms inference latency, indicating suitability for fast contingency screening after offline training. We also observe that retrieval augmentation and error-feedback retries are important for reliable simulation automation, while very high renewable penetration introduces solver/initialization challenges that lower scenario convergence.

The framework offers key advantages: (i) reduced manual scripting through natural-language scenario specification; (ii) automated architecture search eliminating manual trial-and-error; (iii) consistent performance across multiple random seeds; and (iv) scalability to larger systems (96.66% accuracy on IEEE 118-bus). Limitations include dependence on knowledge base quality and reduced simulator convergence for high renewable penetration. Importantly, the framework does not remove expert involvement, but shifts it to an upfront setup of templates, knowledge sources, and safety constraints, after which routine studies can be accelerated. Practical deployment requires engineering validation and human oversight, with the LLM serving as a decision-support tool. Future work will focus on renewable-aware templates, physics-informed search constraints, and field-oriented validation.

Acknowledgment

This paper utilized ChatGPT-5 for data processing and text refinement.

References

[1] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.

[2] R. Yan, T. K. Saha, F. Bai, and H. Gu, "The anatomy of the 2016 south australia blackout: A catastrophic event in a high renewable network," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2727–2742, 2018.

[3] Expert Panel on the Grid Incident on 28 April 2025, "Grid incident in spain and portugal on 28 april 2025," Expert Panel on the Grid Incident on 28 April 2025, Tech. Rep., 2025.

[4] F. Milano, F. D"orfler, G. Hug, D. J. Hill, and G. Verbič, "Foundations and challenges of low-inertia systems," in *Power Systems Computation Conference (PSCC)*. IEEE, 2018, pp. 1–25.

[5] A. A. Fouad and V. Vittal, *Power System Transient Stability Analysis Using the Transient Energy Function Method*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[6] S. Zhang, Z. Zhu, and Y. Li, "A critical review of data-driven transient stability assessment of power systems: Principles, prospects and challenges," *Energies*, vol. 14, no. 21, p. 7238, 2021.

[7] Y. Zhang, Y. Xu, Z. Y. Dong, and R. Zhang, "Power system stability assessment using machine learning: A comprehensive review," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3310–3326, 2021.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24 824–24 837.

[12] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.

979-8-3315-8138-1