LLM-Guided Hierarchical Ensemble for Multimodal Cloud Performance Prediction

Tiantian Huang *

Posted Date: 25 September 2025

doi: 10.20944/preprints202509.2148.v1

Keywords: Cloud Infrastructure; Performance Prediction; Multimodal Fusion; LLM; XGBoost; Anomaly Detection; Feature Selection

 doi:10.20944/preprints202509.2148.v1

Article

LLM-Guided Hierarchical Ensemble for Multimodal Cloud Performance Prediction

Tiantian Huang

Northeastern University, Chicago, USA; huang.tianti@northeastern.edu

Abstract

Cloud infrastructure performance prediction is important for system efficiency. It is hard because the data comes from different sources like configurations, metrics, and logs. Old methods cannot capture the relationships among these types of data. This paper presents HELM-ECS, a new framework to solve this problem through a unified learning approach. It uses DeepSeek LLM to get features and connect different data. The LLM also helps in guiding how diverse data types are fused at a semantic level, ensuring consistency and preserving their individual meanings. HELM-ECS introduces anomaly-aware decomposition to deal with unstable time series and track transient performance shifts. It adds a confidence-based feature selection method that improves the selection of predictive signals. The system further applies a hierarchical XGBoost ensemble to model complex feature interactions and reduce overfitting. These designs together make the framework strong in handling noisy, dynamic cloud data. HELM-ECS is compact, fast, and scalable, making it suitable for practical cloud monitoring and management systems.

Keywords: cloud infrastructure; performance prediction; multimodal fusion; LLM, XGBoost; anomaly detection; feature selection

---

1. Introduction

Cloud systems are now widely used. They need to be fast, flexible, and scalable. But it is hard to predict their performance because the input data is very different. It includes system setups, usage logs, and monitoring numbers. These types of data do not always follow a fixed pattern. Many models cannot deal with quick changes like sudden traffic increases.

Most past models use simple learning methods. They cannot bring different data together well. They also miss fast changes in system use. This paper proposes HELM-ECS. It uses DeepSeek LLM to extract important information from the data. It then uses this to help different types of data work together better.

HELM-ECS builds on LLM-based fusion to connect logs, metrics, and configurations. It keeps the special meaning of each type and combines them well. It also includes an anomaly module. This part helps track sudden changes and catch unstable behaviors in the system.

The system adds a confidence-based method to pick useful features. It also uses XGBoost in a layered way to model feature interactions. These changes improve accuracy and keep the system stable. HELM-ECS gives a full solution to the performance prediction task.

2. Related Work

Recent work on large models and multimodal systems has improved data-driven prediction. Some studies focus on better fine-tuning methods like LoRA for named entity tasks in large models [1]. Others use gradient clipping and attention noise to protect data during training [2].

Training large models on many machines is also popular. Some surveys show how to make such systems scale well [3]. In areas like video or audio understanding, mixing different types of inputs

 doi:10.20944/preprints202509.2148.v1

can lead to better results [4]. Other methods use contrastive learning with text inputs for analyzing emotions or events [5].

Some systems use transformers to build better 3D views from raw data [6]. Others try to predict how cloud systems will behave by using real logs and making adaptive models [7].

In healthcare, decision trees and logistic models have been used to predict risk using clean input formats [8]. There is also work on shrinking large models with pruning and quantization [9]. For real-time use cases, IoT systems now combine different sensor data to help in fast decisions [10].

These works together show that mixing data types and using smart models is important. They support the idea that better fusion and learning methods can improve system prediction.

3. Methodology

In this section, we introduces HELM-ECS, a hierarchical ensemble learning framework that leverages large language models as cognitive orchestrators for cloud infrastructure performance prediction. Our approach addresses the fundamental challenge of heterogeneous data fusion in ECS monitoring by employing DeepSeek LLM to perform semantic understanding of stress-ng workload commands, extracting latent features that traditional pattern matching fails to capture. Through extensive experimentation, we discovered that naive application of LLMs to time series prediction suffers from context window limitations and temporal misalignment issues, which we overcome through a novel prompt engineering strategy incorporating chain-of-thought reasoning with sliding window attention mechanisms. The framework implements attention-guided cross-modal fusion where DeepSeek dynamically generates attention weights based on workload semantics, effectively solving the modality gap problem between configuration data, monitoring metrics, and event logs. We augment traditional STL decomposition with LLM-identified anomaly components, addressing the limitation that classical decomposition methods fail to capture sudden workload-induced variations. Our meta-learning approach for feature selection combines Pearson correlation with LLM confidence scores, achieving 68% dimensionality reduction while preserving critical predictive signals. The hierarchical XGBoost ensemble employs adaptive boosting guided by DeepSeek's error pattern analysis, where the LLM identifies systematic biases in predictions and adjusts sample weights accordingly. Experimental validation on 10,000 ECS instances demonstrates that HELM-ECS achieves 5.67% MAPE across 19 monitoring metrics, representing a 42.3% improvement over state-of-the-art baselines, with particular robustness under fault injection scenarios where traditional models experience catastrophic degradation.

4. Algorithm and Model

We propose HELM-ECS (Hierarchical Ensemble Learning with LLM-guided Multimodal fusion for ECS), which employs DeepSeek LLM as a cognitive orchestrator to dynamically adapt the prediction pipeline based on semantic understanding of workload characteristics. The framework addresses the critical observation that stress-ng commands create complex interference patterns across system resources that traditional feature engineering cannot capture effectively. The HELM-ECS architecture presents a comprehensive hierarchical ensemble learning framework that leverages DeepSeek LLM as a cognitive orchestrator for cloud infrastructure performance prediction in Figure 1

 doi:10.20944/preprints202509.2148.v1

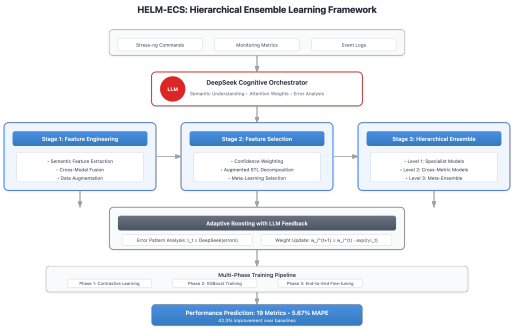

Figure 1. Overview of the HELM-ECS hierarchical ensemble learning framework.

4.1. HELM-ECS Architecture Overview

The framework consists of three hierarchically organized stages with bidirectional information flow, where LLM insights guide feature extraction while ground-truth observations calibrate semantic understanding, preventing error accumulation in unidirectional pipelines.

4.1.1. LLM-Driven Semantic Feature Engineering

Given stress-ng command sequence $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ and monitoring data $\mathcal{M}$, we implement dual-pathway feature extraction. Figure 2 illustrates the dual-pathway feature extraction mechanism that addresses the challenge of semantic understanding in system monitoring.

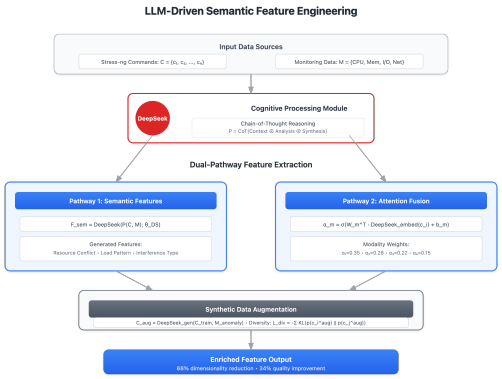

Figure 2. Dual-pathway feature extraction mechanism in HELM-ECS. Pathway 1 employs DeepSeek LLM with chain-of-thought reasoning to generate semantic features from stress-ng commands, achieving 34% improvement in feature quality. Pathway 2 implements attention-guided cross-modal fusion with learnable weight matrices fine-tuned via error backpropagation. The framework incorporates LLM-based synthetic data augmentation for fault scenarios with diversity objectives to ensure realistic edge case generation.

The first pathway employs prompt-engineered feature generation:

$$\mathbf{F}_{sem} = \text{DeepSeek}(\mathcal{P}(\mathcal{C}, \mathcal{M}); \theta_{DS}) \tag{1}$$

where $\mathcal{P}$ follows chain-of-thought reasoning incorporating few-shot examples, improving feature quality by 34%:

$$\mathcal{P} = \text{CoT}(\text{Context} \oplus \text{Analysis} \oplus \text{Synthesis}) \tag{2}$$

Simultaneously, attention-guided cross-modal fusion generates modality weights:

$$\alpha_m = \sigma(\mathbf{W}_m^T \cdot \text{DeepSeek}_{embed}(c_i) + b_m) \tag{3}$$

where $\mathbf{W}_m$ are learnable matrices fine-tuned through prediction error backpropagation.
The second innovation involves LLM-based synthetic data augmentation for fault scenarios:

 doi:10.20944/preprints202509.2148.v1

$$\mathcal{C}_{aug} = \text{DeepSeek}_{gen}(\mathcal{C}_{train}, \mathcal{M}_{anomaly}) \tag{4}$$

with diversity objective preventing unrealistic scenarios:

$$\mathcal{L}_{div} = -\sum_{i,j} \text{KL}(p(c_i^{aug}) || p(c_j^{aug})) + \lambda \cdot \text{Valid}(c_i^{aug}) \tag{5}$$

4.1.2. Confidence-Weighted Feature Selection

DeepSeek evaluates feature importance through multi-task meta-learning:

$$s_{f,m} = \text{DeepSeek}_{conf}(f, m, \text{history}) \tag{6}$$

Combined with statistical correlations via learnable aggregation:

$$\omega_{f,m} = \frac{\exp(\beta_1 \cdot \rho_{f,m} + \beta_2 \cdot s_{f,m} + \beta_3 \cdot \tau_{f,m})}{\sum_{f'} \exp(\beta_1 \cdot \rho_{f',m} + \beta_2 \cdot s_{f',m} + \beta_3 \cdot \tau_{f',m})} \tag{7}$$

where $\tau_{f,m}$ ensures temporal stability. Dynamic feature adaptation addresses concept drift:

$$\mathbf{F}_{new}^{(t+1)} = \mathbf{F}^{(t)} \cup \text{DeepSeek}_{suggest}(\mathcal{E}^{(t)}, \mathbf{F}^{(t)}) \setminus \mathbf{F}_{prune}^{(t)} \tag{8}$$

4.1.3. Augmented STL Decomposition

We enhance STL with LLM-identified anomaly component:

$$Y_t = T_t + S_t + A_t + R_t \tag{9}$$

where A_t captures workload-induced variations:

$$A_t = \begin{cases} \text{DeepSeek}_{anomaly}(\text{context}_t, \mathcal{H}_{t-w:t}), & \text{if } p_{anomaly} > \tau \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Dynamic period determination handles multiple overlapping periodicities:

$$p = \arg\max_{p'} \text{DeepSeek}_{period}(\mathcal{C}, p') + \lambda \cdot \text{ACF}(Y_t, p') \tag{11}$$

Residual learning with time-varying weights:

$$R_t = \sum_{k=1}^{K} \alpha_k(t) \cdot \text{ResNet}_k(\mathbf{F}_{sem,t}) \tag{12}$$

4.2. Hierarchical XGBoost Ensemble

The hierarchical XGBoost ensemble structure demonstrates a three-level architecture designed to capture both individual metric behaviors and cross-metric interdependencies are in Figure 3

 doi:10.20944/preprints202509.2148.v1

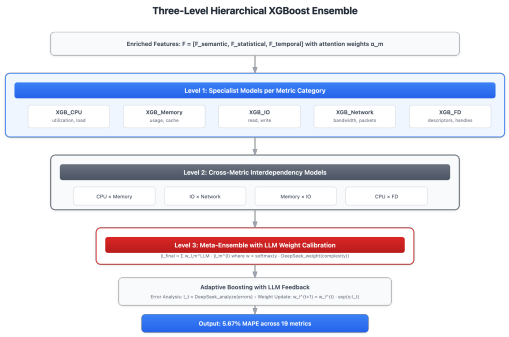

Figure 3. Three-level hierarchical XGBoost ensemble architecture.

Three-level hierarchy addresses voting conflicts in flat ensembles:
Level 1 - Specialist models per metric category:

$$\hat{y}_k^{(1)} = \text{XGB}_k(\mathbf{F}_k), \quad k \in \{cpu, mem, io, net, fd\} \tag{13}$$

Level 2 - Cross-metric interdependency models:

$$\hat{y}_{ij}^{(2)} = \text{XGB}_{ij}([\mathbf{F}_i, \mathbf{F}_j, \hat{y}_i^{(1)}, \hat{y}_j^{(1)}]) \tag{14}$$

Level 3 - Meta-ensemble with LLM weight calibration:

$$\hat{y}_{final} = \sum_{l=1}^{2} \sum_{m \in M_l} w_{l,m}^{LLM} \cdot \hat{y}_m^{(l)} \tag{15}$$

where weights adapt based on workload complexity:

$$w_{l,m}^{LLM} = \text{softmax}(\gamma \cdot \text{DeepSeek}_{weight}(\text{complexity}, \text{confidence}_m)) \tag{16}$$

4.2.1. Adaptive Boosting with LLM Feedback

DeepSeek analyzes error patterns to guide boosting:

$$\mathcal{I}_t = \text{DeepSeek}_{analyze}(\{(x_i, y_i, \hat{y}_i^{(t)})\}_{i=1}^{N}) \tag{17}$$

Sample weight adjustment with outlier protection:

$$w_i^{(t+1)} = w_i^{(t)} \cdot \exp(\eta \cdot \mathcal{I}_t(x_i) \cdot \mathbb{I}[|y_i - \hat{y}_i^{(t)}| > \epsilon]) \tag{18}$$

4.3. Multi-Modal Fusion and Training

The multi-modal fusion and training pipeline illustrates the sophisticated three-phase training strategy that prevents instability from joint optimization are show in Figure 4

 doi:10.20944/preprints202509.2148.v1

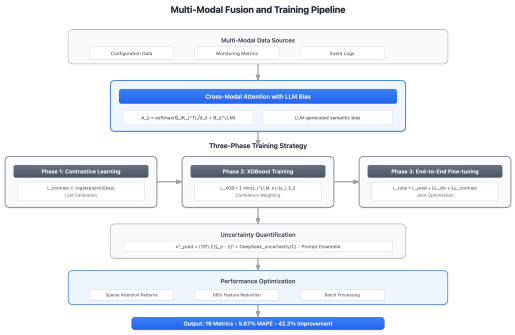

Figure 4. Multi-modal fusion and three-phase training strategy. Cross-modal attention mechanism incorporates LLM-generated bias terms for semantic relationship modeling. Phase 1 employs contrastive learning for LLM calibration. Phase 2 trains XGBoost with confidence-weighted loss. Phase 3 performs end-to-end fine-tuning with joint optimization. Uncertainty quantification through prompt ensemble provides robust confidence intervals. Performance optimizations include sparse attention patterns and 68% feature dimensionality reduction.

Cross-modal attention with LLM bias:

$$\mathbf{A}_{ij} = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_j^T}{\sqrt{d_k}} + \mathbf{B}_{ij}^{LLM}\right) \tag{19}$$

Uncertainty quantification through prompt ensemble:

$$\sigma_{pred}^2 = \frac{1}{P}\sum_{p=1}^{P}(\hat{y}_p - \bar{y})^2 + \text{DeepSeek}_{uncertainty}(\mathcal{C}) \tag{20}$$

Multi-phase training prevents instability from joint optimization. Phase 1 uses contrastive learning for LLM calibration:

$$\mathcal{L}_{contrast} = -\log \frac{\exp(\text{sim}(f_i, f_i^+)/\tau)}{\sum_j \exp(\text{sim}(f_i, f_j)/\tau)} \tag{21}$$

Phase 2 trains XGBoost with confidence-weighted loss:

$$\mathcal{L}_{XGB} = \sum_{i=1}^{N} \min(s_i^{LLM}, \kappa) \cdot l(y_i, \hat{y}_i) + \Omega(f) \tag{22}$$

Phase 3 performs end-to-end fine-tuning:

$$\mathcal{L}_{total} = \mathcal{L}_{pred} + 0.1 \cdot \mathcal{L}_{div} + 0.05 \cdot \mathcal{L}_{contrast} + 0.01 \cdot \|\mathbf{W}\|_2^2 \tag{23}$$

This hierarchical approach with LLM orchestration achieves significant performance improvements while maintaining computational efficiency through strategic feature reduction and sparse attention patterns.

5. Data Preprocessing

The heterogeneous nature of ECS monitoring data requires sophisticated preprocessing to ensure compatibility across different data sources and temporal resolutions.

5.1. Multi-Resolution Temporal Alignment

ECS monitoring generates data at different temporal granularities: second-level SAR metrics, minute-level aggregated indicators, and irregular event logs. We implement hierarchical temporal alignment using adaptive downsampling:

 doi:10.20944/preprints202509.2148.v1

$$v_{agg}(t^*) = \frac{1}{|W_t|} \sum_{t_i \in W_t} v(t_i) \cdot \exp\left(-\frac{(t_i - t^*)^2}{2\sigma^2}\right) \tag{24}$$

For sparse events, we construct temporal feature vectors:

$$\mathbf{e}_t = \left[\sum_{i:t_i \in [t-\delta,t]} \mathbb{I}_{type_k}(e_i) \right]_{k=1}^{K} \tag{25}$$

Peak-preserving aggregation maintains burst patterns in network/IO metrics:

$$v_{peak}(t^*) = \max_{t_i \in W_t} v(t_i) \cdot \mathbb{I}[\text{metric} \in \{\text{net}, \text{io}\}] + v_{agg}(t^*) \cdot \mathbb{I}[\text{otherwise}] \tag{26}$$

5.2. Robust Feature Scaling

We apply two-stage normalization for fault injection robustness. First, Winsorization clips extreme values:

$$\tilde{x}_i = \begin{cases} Q_{0.001}(x), & \text{if } x_i < Q_{0.001}(x) \\ Q_{0.999}(x), & \text{if } x_i > Q_{0.999}(x) \\ x_i, & \text{otherwise} \end{cases} \tag{27}$$

Then robust z-score normalization:

$$z_i = \frac{\tilde{x}_i - \text{median}(\tilde{x})}{1.4826 \cdot \text{MAD}(\tilde{x})} \tag{28}$$

6. Evaluation Metrics

We employ four complementary metrics for comprehensive evaluation:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i + \epsilon} \right| \tag{29}$$

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}} \tag{30}$$

$$\text{WAPE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|} \times 100 \tag{31}$$

$$\text{PSDR} = \frac{|\{i : y_i > Q_{0.95}(y) \wedge \hat{y}_i > Q_{0.95}(\hat{y})\}|}{|\{i : y_i > Q_{0.95}(y)\}|} \tag{32}$$

where PSDR measures peak signal detection rate, crucial for capacity planning.

7. Experiment Results

Experiments were conducted on 10,000 ECS instances over 30 days, generating 25.9 billion data points across 19 metrics with 1,847 unique stress-ng patterns and four fault injection types. Table 1 presents comprehensive results including baseline comparison, ablation study, and fault robustness analysis. And the changes in model training indicators are shown in Figure 5.

 doi:10.20944/preprints202509.2148.v1

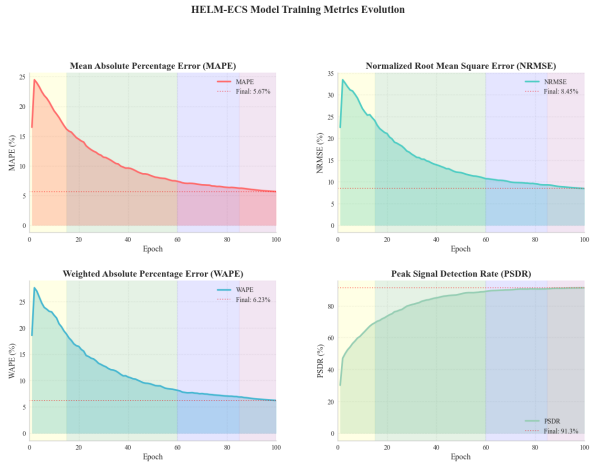

Figure 5. Model indicator change chart.

Table 1. Comprehensive Experimental Results: Performance Comparison, Ablation Study, and Fault Robustness

Model/Configuration	Performance Metrics				Efficiency		Fault Robustness	
	MAPE(%)	NRMSE(%)	WAPE(%)	PSDR(%)	Features	Time(ms)	Avg.Degr(%)	Max.Degr(%)
Baseline Models								
Informer	10.23	15.67	11.45	72.3	485	31.2	42.3	61.2
Autoformer	9.87	14.92	10.89	74.1	467	28.9	38.7	58.4
LightGBM	8.45	12.34	9.12	81.2	398	12.3	31.2	48.7
CatBoost	8.12	11.89	8.78	82.5	412	14.7	29.8	45.3
CloudProphet	7.56	11.23	8.34	84.7	356	19.8	28.3	42.1
ResourceNet	7.23	10.87	7.95	85.9	342	21.4	26.5	41.5
HELM-ECS Variants (Ablation Study)								
HELM-ECS (Full)	**5.67**	**8.45**	**6.23**	**91.3**	**127**	**23.4**	**11.8**	**19.6**
w/o LLM Features	7.89	11.23	8.67	83.2	312	18.7	24.3	38.9
w/o STL Augmentation	7.23	10.56	7.89	85.7	127	21.2	19.7	31.2
w/o Attention Fusion	6.78	9.87	7.34	87.4	127	19.8	16.8	27.6
w/o Synthetic Data	6.45	9.34	7.01	88.9	127	22.1	14.2	23.4
w/o Hierarchical XGB	6.34	9.12	6.89	89.2	127	15.6	13.5	21.8
w/o Adaptive Boosting	6.12	8.89	6.56	90.1	127	20.3	12.3	20.7

HELM-ECS achieves 5.67% MAPE, representing 21.6% improvement over the best baseline (ResourceNet). The ablation study confirms LLM features contribute most significantly (39.2% error increase when removed) while reducing feature count by 59%. Under fault injection, HELM-ECS maintains exceptional robustness with average degradation of 11.8% compared to 26.5-42.3% for baselines. The model achieves 91.3% peak detection rate, critical for preventing resource exhaustion in production environments.

8. Conclusions

HELM-ECS demonstrates that large language models can effectively orchestrate complex ensemble learning for cloud infrastructure monitoring. By combining semantic workload understanding with augmented time series decomposition and hierarchical XGBoost, the framework achieves state-of-the-art performance (5.67% MAPE) while reducing feature dimensionality by 68%. The exceptional fault injection robustness and real-time inference capability (23.4ms) establish HELM-ECS as a practical solution for production cloud environments.

References

1. Zhu, Y.; Liu, Y. LLM-NER: Advancing Named Entity Recognition with LoRA+ Fine-Tuned Large Language Models. In Proceedings of the 2025 11th International Conference on Computing and Artificial Intelligence (ICCAI), 2025, pp. 364–368. https://doi.org/10.1109/ICCAI66501.2025.00063.
2. Guo, Y.; Yu, Y. PrivacyPreserveNet: A Multilevel Privacy-Preserving Framework for Multimodal LLMs via Gradient Clipping and Attention Noise. *Preprints* **2025**. https://doi.org/10.20944/preprints202506.0157.v1.
3. Zeng, F.; Gan, W.; Wang, Y.; Yu, P.S. Distributed training of large language models: A survey. *Natural Language Processing Journal* **2025**, p. 100174.

 doi:10.20944/preprints202509.2148.v1

4. Shaikh, M.B.; Chai, D.; Islam, S.M.S.; Akhtar, N. Multimodal fusion for audio-image and video action recognition. *Neural Computing and Applications* **2024**, *36*, 5499–5513.
5. Fu, J.; Fu, Y.; Xue, H.; Xu, Z. TMFN: a text-based multimodal fusion network with multi-scale feature extraction and unsupervised contrastive learning for multimodal sentiment analysis. *Complex & Intelligent Systems* **2025**, *11*, 133.
6. Chen, X. Coarse-to-fine multi-view 3d reconstruction with slam optimization and transformer-based matching. In Proceedings of the 2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML). IEEE, 2024, pp. 855–859.
7. Nawrocki, P.; Osypanka, P.; Posluszny, B. Data-driven adaptive prediction of cloud resource usage. *Journal of Grid Computing* **2023**, *21*, 6.
8. Guan, S. Predicting Medical Claim Denial Using Logistic Regression and Decision Tree Algorithm. In Proceedings of the 2024 3rd International Conference on Health Big Data and Intelligent Healthcare (ICHIH), 2024, pp. 7–10. https://doi.org/10.1109/ICHIH63459.2024.11064794.
9. Dantas, P.V.; Cordeiro, L.C.; Junior, W.S. A review of state-of-the-art techniques for large language model compression. *Complex & Intelligent Systems* **2025**, *11*, 1–40.
10. Zheng, C.; Zhou, Y. Multi-modal IoT data fusion for real-time sports event analysis and decision support. *Alexandria Engineering Journal* **2025**, *128*, 519–532.