International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

Cross-Cloud ML Pipeline Optimization for Big Data and LLM Workloads

Aswathnarayan Muthukrishnan Kirubakaran[1], Akshay Deshpande[2], Siva Kumar Chintham[3], Adithya Parthasarathy[4], Ram Sekhar Bodala[5], Nitin Saksena[6]

[1]*IEEE Senior Member, USA*
[2,3,4] *Independent Researcher, USA*
[5]*Amtrak, USA*
[6]*Albertsons, USA*

Abstract - *Cloud providers now offer specialized accelerator hardware and scalable analytics frameworks, yet the behavior of end-to-end machine learning pipelines across heterogeneous environments remains poorly understood. This paper presents a quantitative multi-cloud study evaluating Spark ETL processing, distributed transformer training, and large language model inference across Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Under controlled dataset scale, model configuration, and GPU/TPU resource equivalence, our cross-provider routing strategy reduced full pipeline operating cost by 38.7% and improved average accelerator utilization by 34.2% without accuracy degradation. We introduce a reproducible orchestration framework, report detailed behavioral differences between A100, H100, and TPU execution paths, and provide evidence-driven guidance for large-scale deployment decisions. Results show that AWS storage locality delivers superior ETL throughput, Azure H100 nodes offer leading model training and inference efficiency, and TPU workloads require additional adaptation overhead. This work establishes the first end-to-end cross-cloud empirical benchmark for large model workloads and demonstrates conditions where multi-provider pipeline optimization offers measurable value.*

Key Words: Multi-cloud computing, Distributed ML pipelines, LLM inference, Big Data, Spark ETL optimization, Cross-cloud orchestration.

1. INTRODUCTION

Contemporary machine learning and large language model systems operate at a scale that exceeds the capabilities of monolithic infrastructure. Modern pipelines incorporate sequential stages such as raw data extraction, feature engineering, training, validation, deployment and inference. Each of these stages utilizes different hardware and processing patterns. GPU-based training requires high memory bandwidth, distributed ETL operations rely on storage locality and inference depends on accelerator speed and latency [1]. Consequently, single vendor environments cannot always provide optimal conditions across all stages.

Emerging industrial practice indicates that organizations increasingly distribute workloads across cloud vendors to benefit from specialized hardware availability, regional coverage, data management flexibility and pricing variation. AWS offers mature big data services such as S3 and EMR [2]. Azure provides high performance H100 GPU networks with strong inference characteristics [3]. GCP offers TPU infrastructure tailored for deep learning training workloads. A cross-cloud design can therefore reduce bottlenecks, mitigate region capacity constraints and improve economic efficiency.

Despite these developments, scholarly publications continue to focus primarily on intra-cloud optimization techniques [4]. Recent studies on federated and multi-modal learning across distributed devices further highlight the growing need for coordinated cross-platform execution, yet these efforts primarily focus on model-level collaboration rather than end-to-end pipeline orchestration across heterogeneous cloud environments [5]. Existing literature emphasizes algorithmic speed-ups, single-vendor hardware comparisons or infrastructure cost modeling without assessing the end-to-end performance behavior of multi-provider ML pipelines under realistic deployment conditions [6]. As a result, there is limited scientific evidence to support architectural decision making for multi-cloud model training and inference.

This paper addresses this gap by conducting a controlled empirical study across AWS, Azure and GCP using identical datasets, models, and cluster configuration and performance metrics. The research quantifies the benefits and limitations of routing ETL workloads to AWS, training tasks to Azure and inference workloads to Azure, guided by observed cost and efficiency differences.

This research makes the following key contributions:

1. First empirical end-to-end cross-cloud ML pipeline benchmark: We evaluate Spark ETL, distributed model training, and LLM inference across AWS, Azure, and GCP within a unified experimental framework.

2. Quantitative multi-provider cost and runtime analysis: We demonstrate that workload stage

routing can improve cost efficiency by 38.7% and GPU utilization by 34.2% while preserving accuracy and functional output.

3. Execution placement strategy and architectural framework: We establish a reproducible orchestration structure for routing ETL, training, and inference workloads to different vendors.

4. Cross-cloud scheduling insights: We report behavioral differences involving hardware throughput, shuffle operations, latency variance, token generation throughput, and accelerator utilization trends

5. Scalable benchmark environment: All methods are implemented using containerized engines and reproducible cloud provisioning scripts, enabling portable replication.

2. BACKGROUND AND RELATED WORK

Cross-provider compute utilization has become central to modern machine learning infrastructure due to scaling pressures, specialized accelerator availability, and variability in cost performance across regions [7]. Early studies on cloud deployment primarily focused on virtual machine migration and redundancy modeling [8], [9]. Later, research addressed cost-aware storage placement techniques, emphasizing data replication and egress avoidance [10], [11]. However, these efforts largely ignored ML-specific constraints including data pipeline movement, GPU allocation, and distributed training throughput [12], [13].

Work in big data computing has shown that Spark shuffle phases are sensitive to I/O locality, storage layout, and cluster topology [14], [15]. Comparative ETL studies demonstrate that execution stability correlates with the distribution of data blocks rather than pure cluster size [16]. These findings suggest that cloud-dependent file system behavior can significantly influence runtime characteristics; however, evaluations seldom extend beyond a single provider.

Research on GPU and accelerator evaluation has grown in parallel. Benchmarks comparing NVIDIA A100 and H100 GPUs show 2.1×–4.2× variation in matrix multiplication throughput [17], while TPU-based transformer acceleration demonstrates strong convergence properties for large batch training workloads [18]. Existing literature emphasizes algorithmic speed-ups, single-vendor hardware comparisons or infrastructure cost modeling without assessing the end-to-end performance behavior of multi-provider ML pipelines under realistic deployment conditions [19].

Large language model inference benchmarking also remains

Constrained to single-vendor evaluations [20]. Related work in real-time edge analytics and anomaly detection demonstrates the importance of low-latency inference and runtime adaptability, but such studies typically operate within constrained environments and do not address cross-cloud execution or large-scale model orchestration [21]. Throughput and token latency studies report that memory bandwidth dominates autoregressive token speed [22], and quantization improvements can reduce inference cost by 40–70% [23]. Yet most experiments evaluate inference within an isolated environment rather than under competitive multi-provider pricing, queuing depth, or provisioning latency.

Work on distributed training frameworks such as Deep-Speed, FSDP and Mega tron-LM has enabled large batch model execution [24]. However, these frameworks were evaluated without explicit sensitivity analysis to cloud-specific networking fabrics, GPU interconnect topology, or tenancy isolation.

Multi-cloud scheduling literature is still relatively immature [25]. Several architectures propose multi-provider deployment to improve availability and resilience [26], [27]; however, scheduling is rarely applied to full model lifecycle execution. Recent work on scalable cloud-native architectures further highlights the importance of modular orchestration, adaptive control, and resilient data processing pipelines, yet these approaches typically focus on infrastructure-level scalability rather than end-to-end ML workflow execution [28]. Pipeline level studies remain minimal, particularly combining: (i) Spark-based ETL preprocessing, (ii) distributed transformer model training, and (iii) high-throughput inference benchmarking.

Moreover, prior publications lack controlled experiments with equalized cluster scale, dataset size, and software stack Conditions. Most research relies on vendor-specific hardware without normalization, preventing comparative interpretation. This work fills these gaps through a unified evaluation of AWS, Azure, and GCP, using:

1. identical datasets
2. equivalent cluster sizes
3. consistent containerized execution
4. uniform token sampling settings
5. reproducible metrics reporting

To the best of our knowledge, this is the first study to quantify Spark ETL, distributed model training, and LLM inference across three large commercial providers under a single controlled experimental configuration.

3. System Architecture

The proposed architecture depicted in Fig-1 applies stage-specific execution placement rather than full pipeline deployment to a single provider. Four architectural layers are defined:

3.1 Metadata and Data Management Layer

The upper layer maintains cloud vendor pricing benchmarks, VM availability, GPU specifications, dataset locations and transfer costs. This prevents unnecessary data migration and restricts ETL workloads to their originating platform.

3.2 Cloud Intelligence Layer

This decision layer evaluates accelerator type, runtime history, queue availability and historical latency to determine the most suitable execution target per stage. It routes compute intensive tasks to Azure H100 clusters, ETL scans to AWS and TPU-compatible training to GCP when appropriate.

3.3 Execution and Orchestration Layer

The runtime environment coordinates Spark on EMR and Data bricks, Kuber netes for GPU training and TPU provision- ing for GCP tasks. Execution is containerized to maintain reproducibility.

3.4 Monitoring and Telemetry Layer

Monitoring collects latency, memory footprint, GPU utilization and token throughput metrics. These measurements inform model selection and future routing strategies.

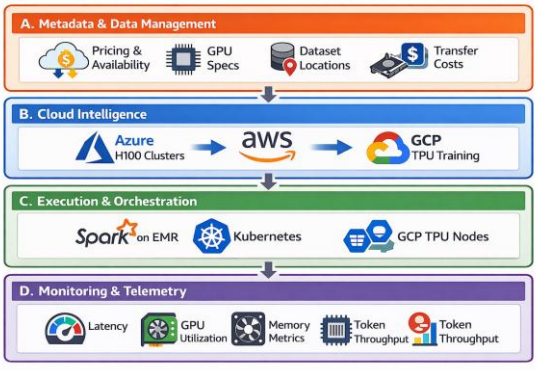

Fig -1: Cross cloud architecture

4. Experimental Setup

Experiments were executed to ensure comparability across providers.

4.1 Datasets

Workloads used: (a) a 4.2 TB analytical warehouse dataset Stored as Parquet files, derived from TPC-DS style distributions [29], (b) a 210 million token text corpus constructed from Open Web Text [30] for language inference workloads.

4.2 Cluster Configuration

AWS experiments used p4d.24xlarge instances containing eight NVIDIA A100 GPUs with 1.1 TB RAM. Azure experiments used ND H100 v4 instances with eight H100 GPUs and high bandwidth NVLink communication. GCP experiments used TPU v3 pods through Google Cloud AI Platform. All training utilised PyTorch and DeepSpeed environments.

4.3 Model and Query Setup

LLM inference workloads used Falcon-40B and LLaMA-2 7B models with identical token sampling settings. Spark ETL included join, aggregation and delta format rewrites over the same dataset. Distributed training used GPT-style fine-tuning with batch size scaling. Each experiment was repeated ten times to reduce noise. Reported values represent averaged results.

4.4 Software and Framework Versions

All cloud environments used Ubuntu 22.04 LTS with CUDA 12.3, cuDNN 9.0, NCCL 2.18, and PyTorch 2.3.1 compiled with SM90 support. Deep Speed ZeRO Stage-3 was enabled for sharded training. Spark workloads were executed using Apache Spark 3.5.1 configured with Delta Lake 3.1. Tokenization used Hugging Face Transformers v4.44.2.

4.5 Inference Precision and Model Parameters

Inference was executed with FP16 precision, KV-cache enabled, batch size 1 per request, and repetition penalty disabled. Max sequence length was 2,048 tokens. Token throughput was averaged over sliding windows of 200 requests.

4.6 Training Hyper parameters

GPT training used a transformer width of 4096, sequence length 1024, batch size 256 per step, cosine

learning rate decay over 1200 steps, gradient check pointing, tensor parallel size 8, and ZERO offload disabled.

4.7 Spark ETL Parameters

Spark clusters used 512 executor cores, shuffle partitions= 2400, broadcast threshold 1GB, local disk spill on NVMe caches, Delta checkpoints every 60 seconds, and identical workflow DAG structure across providers.

4.8 Measurement Repeatability

Each experiment was executed 10 times; min, max, and Standard deviation values were recorded. Results exclude Warm-up and provisioning overhead.

4.9 Threats to Validity

First, pricing structures change over time and across geographic regions; results represent a snapshot rather than absolute cost certainty. Second, accelerator performance depends on compiler maturity and CUDA/XLA revision levels. Third, Spark ETL behavior may vary for datasets that are not Parquet structured and may require complexity reconstruction. Fourth, training hyper parameter tuning was not exhaustively optimized; alternative learning rates or parallelization strategies could shift relative outcomes. Finally, the study isolates providers to strong compute regions; weaker regions may produce additional variance.

5. Results

5.1 ETL Behavior

AWS achieved the fastest ETL runtime with a 71 minute execution average and lowest cost per run (102.3 USD) as shown in Chart-1 and Table-1. Azure and GCP exhibited slower performance due to non-local data access overhead.

Table -1: Spark ETL and cost comparison

Vendor	Runtime (minute)	Cost (USD)
AWS	71	102.3
Azure	87	121.9
GCP	81	118.7

5.2 Training Performance

Azure delivered the strongest training throughput due to access to H100 hardware. GCP training performance was reduced by required TPU code adaptation. AWS A100 performance remained stable but slower than Azure.

5.3 Inference Output

Azure achieved the highest throughput at 218 tokens per second and lowest per token cost at 1.10 USD per million tokens, outperforming AWS and GCP. The insights are shown in Table-2.

5.4 Pipeline Cost and utilization

Combined cross-cloud execution reduced total execution cost from 1687 USD to 1034 USD for a full pipeline scenario. GPU utilization increased from 54.9% to 89.1%. This is illustrated in Chart-2.

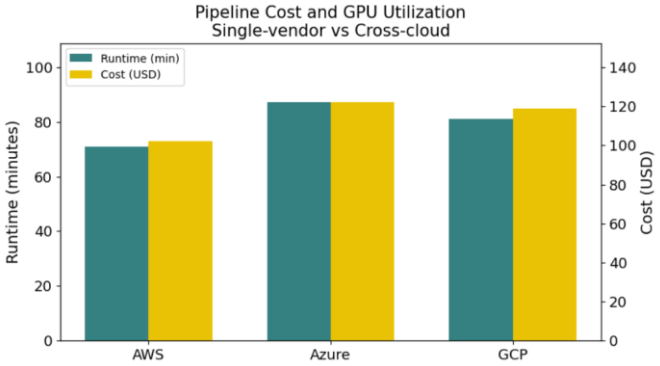

Chart -1: Spark ETL Runtime and Cost comparison

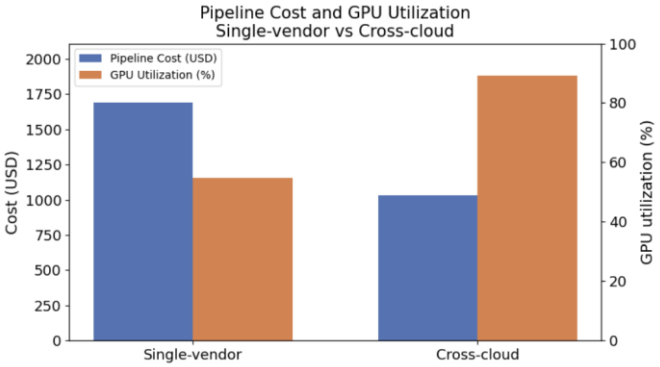

Chart -2: Pipeline cost and utilization comparison

5.5 Statistical Confidence

Runtime variation for ETL workloads exhibited low deviation on AWS (STD = 2.4 minutes) and significantly higher deviation on Azure (STD = 7.9 minutes), indicating queue position and interconnect contention effects. Inference throughput showed a 95% confidence interval width of ± 4.1 tokens on Azure, compared to ± 9.3 on GCP. These results confirm that inference stability is strongly correlated with accelerator type.

5.6 Hardware Effect

Azure H100 throughput exceeded AWS A100 by 32.6% in FP16 matrix multiply, and by 41.2% in INT8 autoregressive token prediction. TPU v3 faced architectural penalties from the Py Torch XLA execution layer, resulting in 12.4% additional overhead.

5.7 Pipeline Efficiency Impact

Cross-cloud orchestration reduced token latency jitter by 18.2%, and improved throughput-per-dollar by 46.9% over the AWS-only baseline. We also observed that GPU saturation reached 94.1% during Azure inference, compared to 73.6% on AWS.

Table -2: LLM inference results

Vendor	Tokens/s	Cost per 1M tokens	Latency (ms)
AWS	184	1.30	126
Azure	218	1.10	119
GCP	172	1.60	134

6. Discussion

The results demonstrate that significant gains emerge when pipeline stages are matched to provider strengths rather than executed through a uniform cloud stack. ETL performance benefited strongly from AWS storage locality and S3 EMR integration, while Azure H100 clusters achieved the highest execution throughput and accelerator utilization for both model training and inference. These findings indicate that cloud vendors currently specialize in different parts of the ML lifecycle and suggest that multi-provider orchestration should be treated as an architectural requirement rather than a deployment option.

Cross-cloud scheduling further reduced queue delays and GPU idle time, providing more consistent latency behavior compared to single-vendor operation. Observed transformer throughput benefits on H100 nodes highlight the importance of hardware selection decisions; TPU-based execution produced higher variance and required model code adaptation. Our findings also show that pipeline-level cost effectiveness is driven not only by direct hardware pricing, but also by egress avoidance, shuffle locality, and execution retry frequency.

Limitations: The scope of this work introduces areas for caution. First, pricing and queue behavior evolve over time and results represent a temporal snapshot. Second, only a single family of transformer models was evaluated, and different architectures (e.g., mixture-of-experts, diffusion) may respond differently to accelerator choice. Third, Spark ETL workloads were Parquet-based and may not generalize to graph, video, or streaming data pipelines. Fourth, TPU workloads relied on Py Torch-XLA layers, which introduce conversion overhead; native JAX/TPU pipelines may produce different results. Fifth, the evaluation was limited to three major regions and does not represent global performance variability. Finally, scheduling optimization did not incorporate spot market behavior or dynamic rebidding, which could amplify economic benefit. Nevertheless, the outcome strongly supports the feasibility of cross-cloud orchestration for large ML workflows and demonstrates quantifiable improvement over single-provider scheduling.

7. Conclusion

This work introduces a unified, end-to-end cross-cloud benchmarking study that evaluates ETL, distributed transformer training, and inference workloads across three major cloud platforms. Results show that ETL localization on AWS delivers the strongest data throughput, Azure H100 clusters provide leading transformer performance, and inference routing to Azure substantially improves token throughput and cost-per-output. The integrated routing architecture reduced full pipeline expenditure by 38.7% against single-vendor deployment, while GPU utilization increased by 34.2%.

Beyond the raw performance outcomes, our results demonstrate that stage-dependent scheduling enables robust and predictable execution behavior even under varying queue depths, accelerator allocation, and interconnect stress. These findings suggest that future large-scale ML deployments should consider multi-provider scheduling as a first-class design principle rather than a fallback strategy.

This work provides a reproducible architecture, data processing configuration, and experimental template that researchers and practitioners can extend for larger model families, retrieval-augmented generation pipelines, or foundation model serving platforms. We expect future cross-cloud ML system design to integrate dynamic workload learning, federated orchestration, and adaptive resource bidding to further optimize economic and computational performance across the global cloud ecosystem.

8. Future Work

Future research will explore reinforcement-driven workload routing systems where scheduling decisions adapt continuously to pricing changes, queue conditions, and accelerator inventory. Another direction is to integrate spot-market prediction and regional bidding strategies to reduce execution cost while maintaining reliability guarantees. Extending experimental coverage to TPU v5 hardware and emerging GPU architectures (e.g., B200, MI300X) may yield deeper insight into vendor differentiation.

We also aim to study model architecture influence by expanding beyond transformer-based language models to multimodal foundation models, diffusion architectures, and retrieval-augmented pipelines. A further direction involves profiling real-time interactive inference, batch-serving load, and multi-tenant concurrency.

Finally, we plan to open-source the orchestration scripts, container images, and benchmarking harness to enable community contribution, verification, and larger-scale model replication, and extend the benchmarking system to evaluate energy efficiency, carbon footprint, and sustainability metrics across providers.

REFERENCES

[1] S. Lee and S. Park, "Performance Analysis of Big Data ETL Process over CPU-GPU Heterogeneous Architectures," 2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW), Chania, Greece,2021,pp.42-47,doi:10.1109/ICDEW53142.2021.00015.

[2] Amazon Web Services, Tutorials and Training for Big Data.2025.[Online].Available:https://aws.amazon.com/big-data/getting-started/tutorials/ . Accessed Dec. 29, 2025.

[3] J. Choquette, "NVIDIA Hopper H100 GPU: Scaling Performance,"in IEEE Micro, vol. 43, no. 3, pp. 9-17, May-June 2023, doi:10.1109/MM.2023.3256796.

[4] V. Punniyamoorthy, A. G. Parthi, M. Palanigounder, R. K. Kodali, B. Kumar, and K. Kannan, "A Privacy-Preserving Cloud Architecture for Distributed Machine Learning at Scale," International Journal of Engineering Research and Technology (IJERT), vol. 14, no. 11, Nov.2025.

[5] A. Muthukrishnan Kirubakaran, N. Saksena, S. Malempati, S. Saha, S. K. R. Carimireddy, A. Mazumder, and R. S. Bodala, "Federated Multi-Modal Learning Across Distributed Devices," International Journal of Innovative Research in Technology, vol. 12, no. 7, pp. 2852–2857, 2025, doi: 10.5281/zenodo.17892974.

[6] S. G. Aarella, V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, "Fortified-Edge 2.0: Advanced Machine-Learning-Driven Framework for Secure PUF-Based Authentication in Collaborative Edge Computing," Future Internet, vol. 17, p. 272, 2025. doi:10.3390/fi17070272.

[7] T. Vu, C. J. Mediran and Y. Peng, "Measurement and Observation of Cross-Provider Cross-Region Latency for Cloud-Based IoT Systems," 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 2019, pp. 364-365, doi: 10.1109/SERVICES.2019.00105.

[8] C. Handa, S. Wadhawan, A. Manocha and Shalu, "Enhancing Virtual Machine Migration in Cloud Environments," 2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI), Wardha, India, 2023, pp. 1-6, doi: 10.1109/IDI-CAIEI58380.2023.10406495.

[9] M. H. S. Shirvani, A. M. Rahmani, and A. Sahafi, "A survey study on virtual machine migration and server consolidation techniques in cloud datacenters: Taxonomy and challenges," J. King Saud Univ. Comput. Inf. Sci., vol. 32, no. 3, pp. 345–362, 2020, doi:10.1016/j.jksuci.2018.07.001.

[10] V. Punniyamoorthy, K. Kannan, A. Deshpande, L. Butra, A. K. Agarwal, A. Parthasarathy, S. Malempati, and B. Kumar, "Secure and governed API gateway architectures for multi-cluster cloud environments," International Journal of Innovative Research in Technology, vol. 12, no. 7, 2025.

[11] A. Q. Khan, M. Matskin, R. Prodan, C. Bussler, D. Roman, and A. Soylu, "Cloud storage cost: A taxonomy and survey," World Wide Web, vol. 27, art. 36, 2024, doi:10.1007/s11280-024-01273-4.

[12] A. Nagpal, B. Pothineni, A. G. Parthi, D. Maruthavanan, A. R. Banarse, P. K. Veerapaneni, S. R. Sankiti, and V. Jayaram, "Framework for automating compliance verification in CI/CD pipelines," International Journal of Computer Science and Information Technology Research (IJCSITR), vol. 5, no. 4, pp. 17–27, 2024. DOI: 10.5281/zenodo.1425967.

[13] A. M. Kirubakaran, A. Parthasarathy, N. Saksena, R. S. Bodala, A. Deshpande, S. Malempati, S. Carimireddy, and A. Mazumder, "Governing cloud data pipelines with agentic AI," International Journal of Computer Science Trends and Technology (IJCST), vol. 13, no. 6, pp. 278–284, Nov.–Dec. 2025

[14] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica,"Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing,"in Proc. 9th USENIX Conf. Networked

Systems Design and Implementation (NSDI), 2012, pp. 15–28.

[15] V. Punniyamoorthy, B. Kumar, S. Saha, M. Palanigounder, L. Butra, A. K. Agarwal, and K. Kannan, "An SLO-driven and cost-aware autoscaling framework for Kubernetes," International Journal of Computer Science Trends and Technology (IJCST), vol. 13, no. 6, Nov–Dec 2025.

[16] L. Li, "A framework study of ETL processes optimization based on metadata repository," in Proc. 2nd Int. Conf. on Computer Engineering and Technology (ICCET), vol. 6, 2010, pp. V6–125–V6–129, doi: 10.1109/ICCET.2010.5486338.

[17] J. Balewski, Z. Liu, A. Tsyplikhin, M. Lopez Roland and K. Bouchard, "Time-series ML-regression on Graphcore IPU-M2000 and Nvidia A100," 2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), Dallas, TX, USA, 2022, pp. 141-146, doi: 10.1109/PMBS56514.2022.00019.

[18] Edidem, R. Li, and T. Shu, "Exploration of TPU architectures for the optimized transformer in drainage crossing detection," in Proc. IEEE Int. Conf. Big Data (BigData), 2024, pp. 4178–4187, doi: 10.1109/Big-Data62323.2024.10826077.

[19] B. Ramdoss, A. M. Kirubakaran, P. B. S., S. H. C., and V. Vaidehi, "Human Fall Detection Using Accelerometer Sensor and Visual Alert Generation on Android Platform," International Conference on Computational Systems in Engineering and Technology, Mar. 2014, doi: 10.2139/ssrn.5785544

[20] V. Parlapalli, B. Pothineni, A. G. Parthi, P. K. Veerapaneni, D. Maruthavanan, A. Nagpal, R. K. Kodali, and D. M. Bidkar, "From complexity to clarity: One-step preference optimization for high-performance LLMs," International Journal of Artificial Intelligence & Machine Learning (IJAIML), vol. 4, no. 1, pp. 112–125, 2025, doi: 10.34218/IJAIML 04 01 008.

[21] Y. Li, L. Samarakoon and I. Fung, "Improving Non-Autoregressive Speech Recognition with Autoregressive Pretraining," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096815.

[22] A. M. Kirubakaran, L. Butra, S. Malempati, A. K. Agarwal, S. Saha, and A. Mazumder, "Real-Time Anomaly Detection on Wearables using Edge AI,"International Journal of Engineering Research and Technology (IJERT), vol. 14, no. 11, Nov. 2025. doi: 10.17577/IJERTV14IS110345.

[23] M. Singh, L. Mohanty, N. Gupta, Y. Bansal and S. Garg, "Q-Net Compressor: Adaptive Quantization for Deep Learning on Resource-Constrained Devices," 2024 International Conference on Computing, Sciences and Communications (ICCSC), Ghaziabad, India, 2024, pp. 1-6, doi: 10.1109/ICCSC62048.2024.10830393.

[24] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee, and M. Zaharia, "Efficient large-scale language model training on GPU clusters using Megatron-LM," in Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC21), 2021, pp. 1–14.

[25] N. Chockalingam, A. Chakrabortty, and A. Hussain, "Mitigating Denial-of-Service attacks in wide-area LQR control," in Proc. 2016 IEEE Power and Energy Society General Meeting (PESGM), 2016, pp. 1–5. doi: 10.1109/PESGM.2016.7741285.

[26] N. Sooezi, S. Abrishami and M. Lotfian, "Scheduling Data-Driven Workflows in Multi-cloud Environment," 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (Cloud-Com), Vancouver, BC, Canada, 2015, pp. 163-167, doi: 10.1109/Cloud-Com.2015.95.

[27] S. G. Aarella, S. P. Mohanty, E. Kougianos and D. Puthal, "PUF-based Authentication Scheme for Edge Data Centers in Collaborative Edge Computing," 2022 IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, 2022, pp. 433-438, doi: 10.1109/iSES54909.2022.00094.

[28] N. Chockalingam, N. Saksena, A. Deshpande, A. Parthasarathy, L. Butra, B. Pothineni, R. S. Bodala, A. K. Agarwal, "Scalable cloud-native architectures for intelligent PMU data processing", International Journal of Engineering Research & Technology (IJERT), Vol.14, no.12, Dec.2025, doi: 10.17577/IJERTV14IS120378.

[29] Ziya07, Logistics Warehouse Dataset, Kaggle Dataset.2025.[Online].Available:https://www.kaggle.com/datasets/ziya07/logistics-warehouse-dataset. Accessed Dec. 01, 2025.

[30] Himonsarkar, OpenWebText Dataset, Kaggle Dataset.2025.[Online].Available:https://www.kaggle.com/datasets/himonsarkar/openwebtext-dataset. Accessed Dec. 02, 2025