Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Explainable AI-Driven Reliability Engineering Framework for Predictive Failure Management and Self-Healing Operations in Hyperscale GPU Clusters

Sudeept Srivastava

Independent Researcher

ARTICLE INFO

Received: 20 June 2026

Revised: 02 Aug 2026

Accepted: 15 Aug 2026

ABSTRACT

The research utilizes explainable AI-based framework of reliability engineering for predictive failure management and self-healing operation to be developed and evaluated in hyper scale GPU clusters. The telemetry data produced were synthetic to model hardware, thermal, fabric, workload and maintenance condition. Random Forest, XGBoost and Random Survival Forest were used with XGBoost performing at an accuracy of 88.12% and an ROC-AUC of 92.13%. SHAP analysis and interpretability, a predictive compute readiness helped with risk-aware intervention. This improved consistency, resilience, efficiency, transparency and operational control is seen in a decrease in interruptions from 258 on the long tail of service metrics to 111, a decrease in MTTR from 5.5 hours to 1.5 hours and an increase in goodput to 99.42% of the long tail.

Keywords: GPU reliability, hyper scale GPU clusters, explainable artificial intelligence, XGBoost, Random Forest, Random Survival Forest, SHAP, predictive maintenance, Predictive Compute Readiness, self-healing infrastructure, autonomous orchestration, operational resilience.

I. INTRODUCTION

Background

Artificial intelligence workloads are becoming dependent on hyper scale GPU clusters, where powerful compute power can be synchronized only within a few microseconds, making infrastructure outages very disruptive [1]. As cluster size increases, there is a greater likelihood that components will fail, causing a decrease in training goodput, an increase in recovery costs and operating reliability becomes compromised. Availability monitoring is thus not enough and predictive health assessment, risk-aware scheduling, automated maintenance and self-healing operations are needed throughout AI infrastructure.

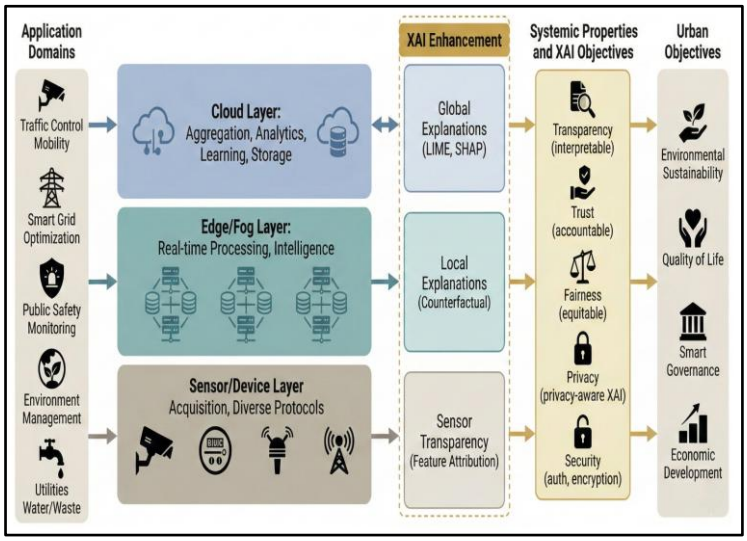

Fig. 1. Next-Gen Explainable AI (XAI) for Federated and Distributed Internet of Things Systems

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Problem Statement

Frequent hardware, interconnect, thermal, firmware, storage, and silent data corruption failures can wreak havoc on tightly coupled AI workloads in hyperscale GPU clusters and result in significant compute resource loss [2]. Executing availability testing on a binary level and then carrying out reactive maintenance to address a failure is not sufficient to predict degradation or degradation impact on the failure [3]. Thus, there is a strong need to develop a reliable framework to incorporate various aspects of explainable prediction, risk-aware scheduling, maintenance automation, and self-healing recovery [4].

Research Aim and Objectives

Aim

The research aims to create and test an explainable AI based reliability model for predicting GPU failure and making self-healing decisions for hyper scale AI computations systems.

Objectives

- **To find failure types and reliability issues impacting hyper scale GPU cluster operations.**
- **To create AI models that can be explained and are capable of predicting the readiness of GPUs and their breakdown risks.**
- **To design the self-healing orchestration for a risk-aware schedule, automated maintenance and validated recovering.**
- **To assess the performance of a framework with regards to prediction, recovery, goodput, reliability and operational efficiency key metrics.**

Novel Contribution

In this research, an explainable closed-loop reliability framework for future GPUs is provided, incorporating both predictive GPU failure modelling and dynamic compute readiness assessment, risk-aware scheduling, automated maintenance and self-healing recovery [5]. It includes evidence of repaired resource reintegration and bounded autonomy, features auditable decision logic that is not seen in anything else, and overcomes limitations of existing conceptual approaches [6]. Empirically evaluated framework to quantify scale metrics such as prediction accuracy, goodput, recovery performance, reliability and operational efficiency is tested [7].

II. RELATED WORK

Reliability Challenges and Failure Patterns in Hyperscale GPU Clusters

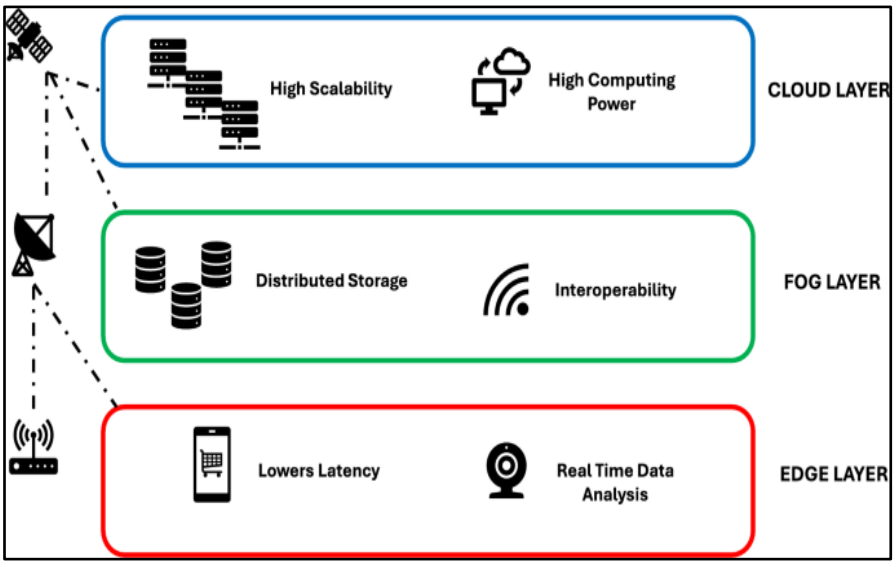

Fig. 2. Taxonomy of Hardware and Network Failure Modes in Large-Scale AI Training Clusters

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Previous literature indicates hyper scale GPU infrastructure does break differently than traditional CPU based infrastructure [8]. High distributed training jobs require tight synchronization between them, where a single fault on an accelerator, memory, fabric, thermal, storage or firmware can have an impact on thousands of healthy devices [9]. A recent example of a system described in Llamas shows this form of exposure to the details of the data in the system when it clearly happens on multiple different types of failures, and in a recent paper from Google and Metamora, it is known that, silent data corruption is shown to achieve this same effect even when the changes do not affect the number of erroneous cells—only the level of accuracy [10]. Unfortunately many of the current evidences are incident dependent and platform-specific, which hinders generalizations to heterogeneous GPUs [11]. The literature thus makes a strong case that failure is inevitable, but does not make a case as strong for how a multiple set of failure measures should be joined together to then measure fleet-level risk [12].

Explainable AI for GPU Failure Risk and Compute Readiness

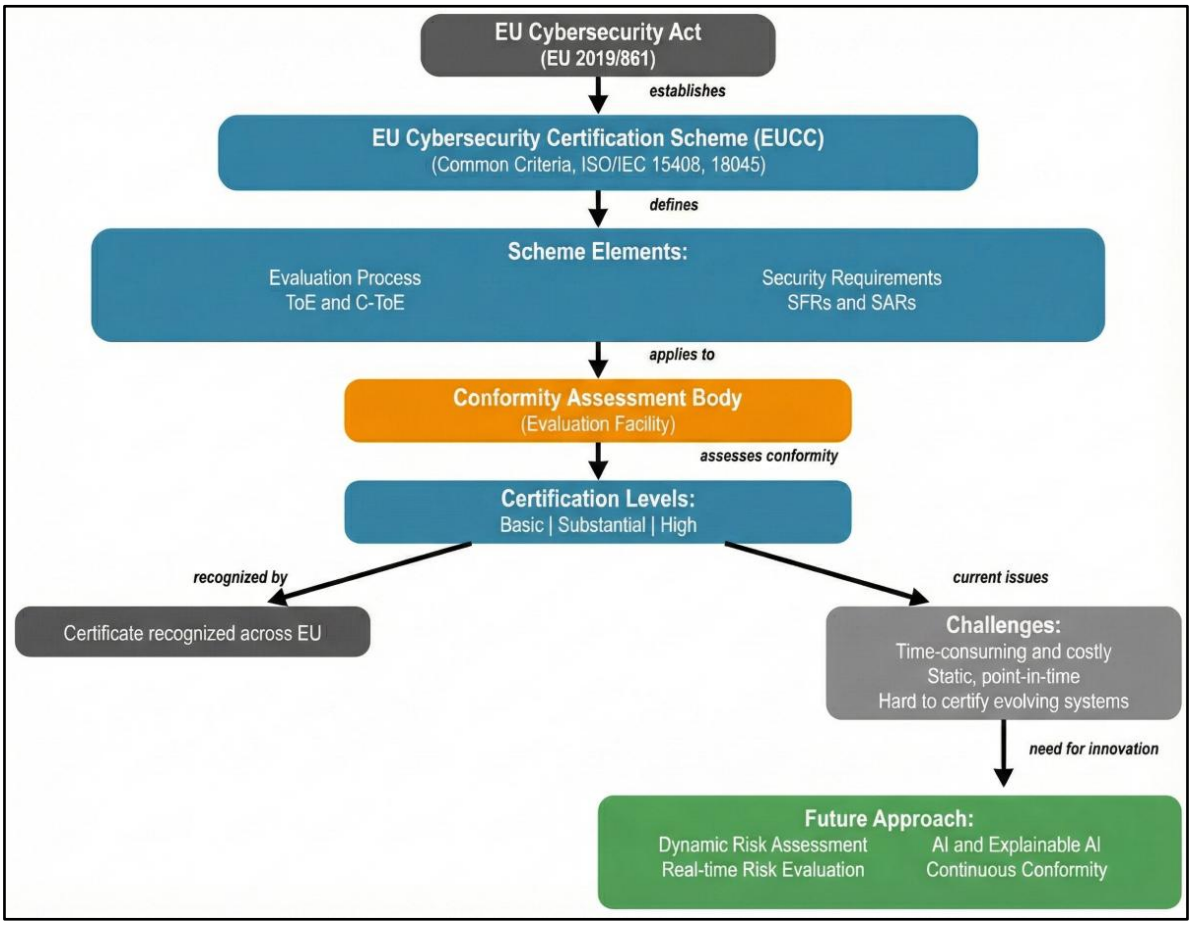

Fig. 3. Explainable AI Framework for GPU Failure Risk and Compute Readiness Mapping

Predictive methods are therefore a more robust option to binary failure monitoring as they predict the chances of failure prior to its occurrence [13]. The engine behind the proposed concept of Predictive Compute Readiness is the ability to integrate layers of hardware, interconnect, topology, workload, maintenance and historic signals to enable more context based scheduling [14]. This is more advanced than classic threshold health checks, however predictive reliability literature has to deal with a critical trade-off – between sensitivity – and operational costs [15]. High false-positive rates can consume healthy GPUs unnecessarily and decrease available capacity [16]. Existing approaches must thus not only be accurate but also specify the reasoning, provide a level of uncertainty and should be interpreted appropriately for the workload [17]. There is a need for explainable AI in that infrastructure operators need to know the reasoning behind a classification of a resource as a risk before allowing automated intervention [18].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Self-Healing Orchestration, Automated Maintenance, and Validated Recovery

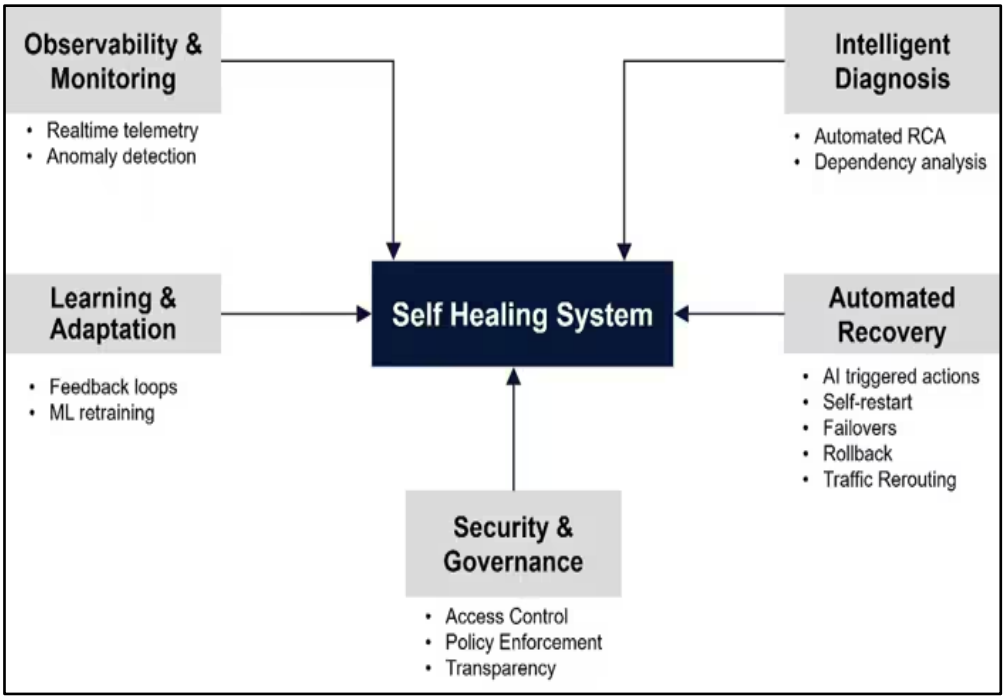

Fig. 4. Self-Healing Systems using AI based Auto-Remediation Strategies

Autonomous infrastructure is a key area of research that increasingly enables closed-loop operations, where telemetry is used to diagnose, remediate and learn [19]. Considering maintenance as a schedule and not as an intromission an external entity, the AI Maintenance Orchestrator expands this logic [20]. Blast-radius prioritization, checkpoint-aligned draining, migration and validation can be used to further minimize service impact over top of the ticket-based approach [21]. However, the full autonomy is not easy to obtain. Unbounded remediation done based on incorrect predictions can lead to remediation outages [22]. The more dominant view in the literature is thus bounded autonomy, which means that automated first, then, if it entails higher risk, evidence, guardrails, and auditability [23]. Preparing things for reintegration is important because repaired hardware can still have existing or recurring faults or may be faulty in a different way [24].

Evaluating Reliability, Recovery, Goodput, and Operational Efficiency

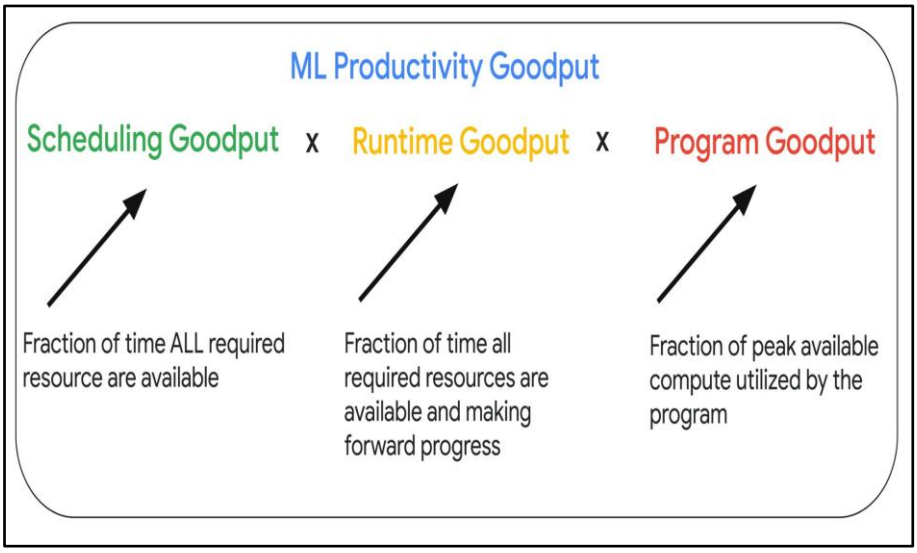

Fig. 5. Performance Benchmarking Pipeline Across Prediction, Recovery, and Cluster Goodput

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A centrally available fleet can be giving sub-optimal useful computation, as measured by the traditional uptime metrics, for large AI clusters [25]. The following dimensions are more meaningful for evaluating the Goodput, prediction precision and recall, interruptions avoided, recovery time, autonomous-action precision and effective cost per useful FLOP [26]. Currently, however, literature does not have common metrics for additively to connect the metrics of prediction, scheduling, maintenance, and recovery [27]. A significant gap exists, however, as existing work has demonstrated useful individual mechanisms, but little empirical study is available to show an explainable and closed-loop framework [28]. It builds on the strengths of predictive readiness, self-healing orchestration, validated recovery and economic performance in a unified way in a single GPU reliability model [29].

Table I: Comparative Analysis Of Ai-Driven Reliability Approaches For Hyperscale Gpu Clusters

Approach	Model Equation	Parameters	Advantages	Limitations	Applicability	Effectiveness
Failure Pattern Analysis	R=f(E,T,F,S,H)	ECC, thermal, fabric, storage, history	Detects degradation	Platform-specific evidence	Fleet monitoring	**High**
Predictive Compute Readiness	PCRi=1−∑wjRij	Risk signals, weights	Risk-aware scheduling	False-positive cost	Failure prediction	**High**
Explainable AI	f(x)=E[f(x)]+∑ϕj	Features, SHAP values	Transparent predictions	Computational overhead	Operator decisions	**High**
Self-Healing Orchestration	A=f(PCR,Risk)	PCR, risk, thresholds	Automated recovery	Automation risk	Maintenance orchestration	**High**
Validated Recovery	RR=AttemptsSuccessful Recovery	Recovery, validation	Prevents faulty reintegration	Validation delay	Post-maintenance control	**High**
Reliability Evaluation	Goodput=Allocated ComputeUseful Compute×100	Compute time, interruptions	Measures useful output	No universal benchmark	Framework evaluation	**Very High**

Literature Gap

Predictions related to these GPU failure patterns, GPUs that are ready to predict, automated maintenance, and self-healing exist, they are still somewhat dispersed in existing researchers. There is very few research that combines

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Explainable Failure Prediction, Risk-aware scheduling, Bounded autonomous remediation and evidence-based recovery within a framework [30]. Furthermore, the absence of standardized measure of the recovery performance, goodput, reliability and operational efficiency combined with an inadequate measure of the measure of prediction accuracy makes it quite clear that integrated validation is needed.

III. RESEARCH METHODOLOGY

Research Design

The research employs quantitative experimental design to design and test a framework for reliability engineering for hyper scale GPU clusters using explainable AI. The methodology takes a closed-loop approach constructed in the same way. Telemetry data is used for prediction, which is then used to schedule and maintain the asset, and when the remediation is executed, the results are labeled feedback to be learned in the future.

Synthetic Data Generation

The simulated data has been created to emulate GPU, node, fabric, workload, and maintenance behavior. Factors that are ECC errors, GPU temperature, power, utilization, memory utilization, NVLink or InfiniBand errors, retransmissions, drift-in firmware, storage contention, topology risk, maintenance history, frequency of checking points, and past failures. The synthetic data set consisted of 4000 observations and 27 variables representing the workload, utilization, thermal, hardware, maintenance, failure, and engineered reliability metrics, it offered enough multidimensionality for later use in a predictive modeling analysis. A binary target will be an indication of whether failure happens at a location within the prediction horizon H:

$$Yf = \{1, \text{Failure occurs within H}\}$$

$$\{0, \text{Otherwise}\}$$

Where $Y_f$represents the failure target and H represents the future prediction horizon.

Time-to-Failure Equation

$$TTF = t_{failure} - t_{observation}$$

Where TTF represents **time-to-failure**, failure is the recorded failure time, and observation is the telemetry observation time.

Data Preprocessing and Feature Engineering

Missing numerical values will be imputed using medians, while categorical values will use modal replacement. Continuous variables will be standardized:

$$Z = \frac{x - \mu}{\sigma}$$

Rolling trends will capture degradation. For example:

$$ECC_{trend} = \frac{ECC_t - ECC_{t-k}}{k}$$

ECCt is the current ECC errors and ECCt−k is previous ECC errors where k is the time interval. Derived features includes thermal margin, failure frequency, maintenance regency, interconnect instability, workload intensity, consistency of the firmware and recovery history. The mean readiness of PCR is 0.719 with a standard deviation of 0.083, it ranges from 0.0413 to 0.919. Consider the generally healthy but varied operating conditions of all GPUs where the median is 0.727.

Predictive Modeling

The following three machine-learning models will be contrasted. A solid nonlinear random forest model for heterogeneous telemetry will be offered by the Random Forest model (HRT). XGBoost can predict complex interactions, rare degradation patterns with boosted decision trees. Random Survival Forest will provide an estimate of the time to failure, which will be useful for maintenance planning beyond just categorization.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Predictive Compute Readiness will integrate normalized risk signals:

$$PCR_i = 1 - \sum_{j=1}^{m} w_j R_{ij}$$

PCRishows readiness score; wj is feature weight, Rij denotes risk value; m is number of risk factors. Where higher PCR indicates greater expected readiness.

Explain ability and Self-Healing

SHAP will explain individual predictions:

$$\mathbf{f(x) = E[f(x)] + j = 1\sum M\phi j}$$

f(x) is model prediction, E[f(x)] denotes baseline prediction, ϕj denotes SHAP contribution.

Predicted risk elicits bounded actions like monitoring, drainage, migration, maintenance, validation, reintegration or quarantine. That's indicative of the source's implementation of progressive autonomy and evidence-based recovery.

Performance Evaluation

The dataset has been split into 80% train and 20% test. Evaluation metrics will be: Precision, Recall, F1 score, ROC-AUC, Prediction calibration and Time to failure accuracy. The approaches used to measure operational effectiveness are mean time to recovery, number of interruptions avoided, number of false predictive drains, and rate of success reintegration, goodput and operational efficiency:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP + FP}{TP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The accuracy is the ratio of correctly predicted failure and non-failure GPU instances to the total number of predictions. True positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are used.

Precision is the percentage of fault predicted GPUs that actually happen to fail. In a situation with imbalanced failure and non-failure cases (diseased, healthy), F1-score is the harmonic mean of precision and recall. ROC-AUC is the model's overall performance at various decision thresholds in distinguishing between failure cases and non-failure cases. TPR is the true positive rate and FPR is the false positive rate.

Operational reliability also is examined with regards to Mean Time to Recovery:

$$\text{MTTR} = \frac{\sum \text{Recovery Time}}{\text{Number of Incidents}}$$

MTTR is the Mean Time to Recovery and refers to the average time needed for the recovery of GPU resources or services following an incident or event. Recovery Time is the length of time from failure up until recovery is successful; Number of Incidents is the total number of failure events recorded.

and training goodput:

$$\text{Goodput} = \frac{\text{Useful Compute Time}}{\text{Total Allocated Compute Time}} \times 100$$

Goodput is the ratio of the total allocated compute time of GPU that is conducted to be useful in computation.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Useful Compute Time represents "productive" GPU time (useful, in this context, meaning the time that is devoted to workloads during the evaluation period), whereas Total Allocated Compute Time represents access to the total capacity of the GPUs assigned to workloads during the evaluation period.

Architecture Diagram

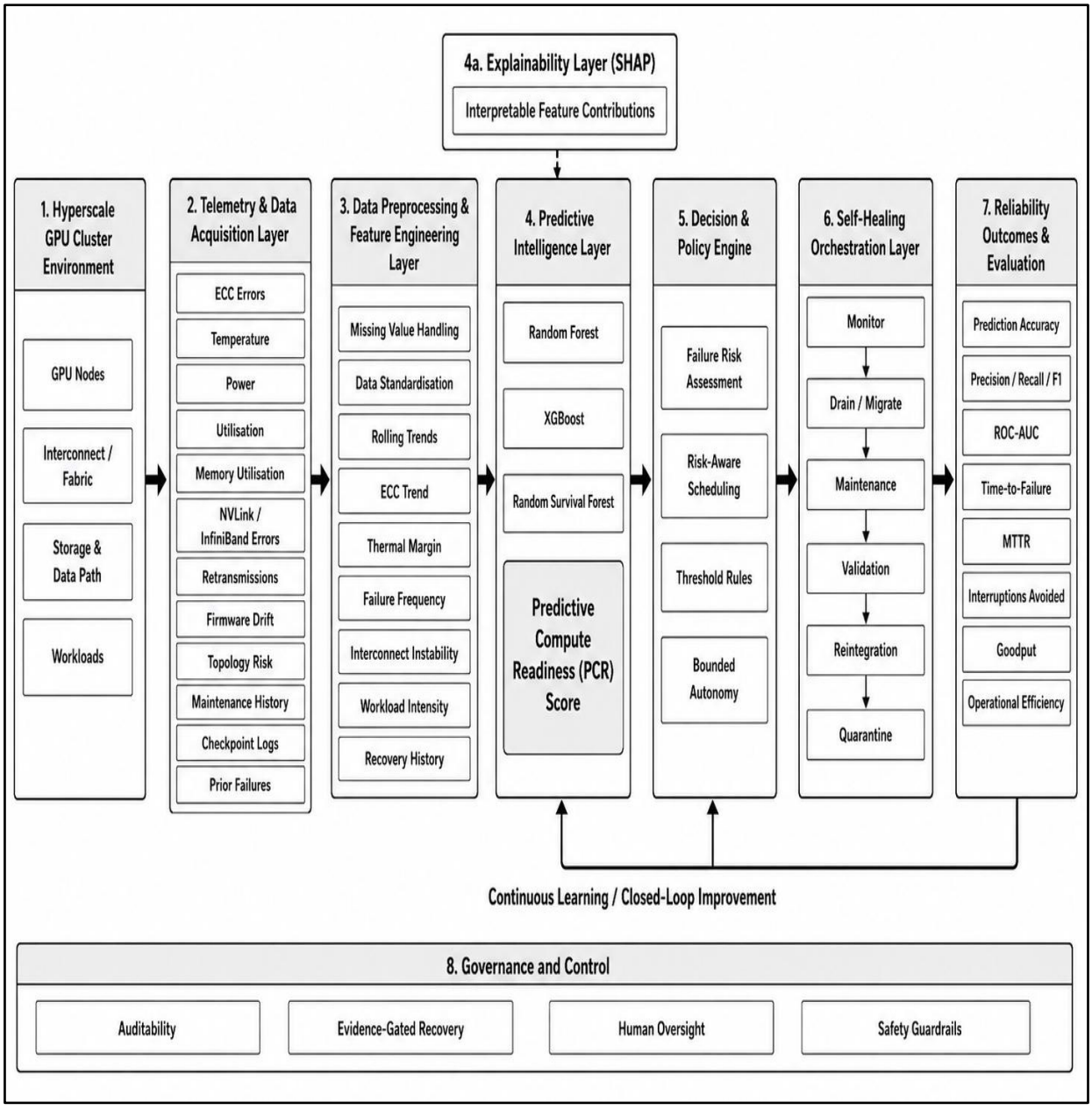

Fig. 6. Architecture Diagram

The architecture combines GPU telemetry, GPU pre-processing, predictive models, PCR scoring, explainability, evaluation and failure interpretation and healing actions into a closed-loop architecture, which allows predicting failures, validated recovery, continuous learnability, and enhanced reliability performance.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Flowchart

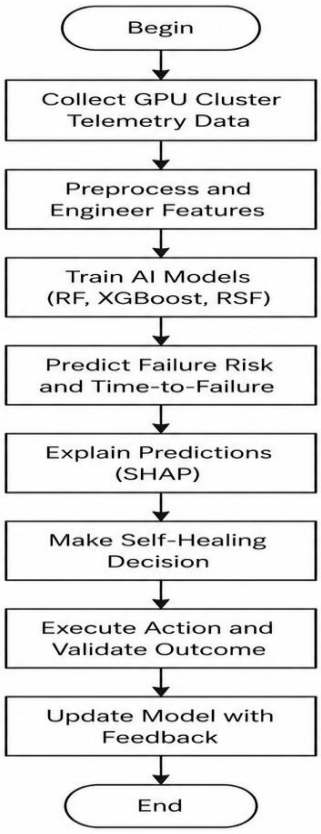

Fig. 7. Flowchart Diagram

The flowchart represents the entire research process, which includes the collection of telemetry data from the GPU, subsequent processing, feature engineering, predictive modeling, explain ability, policy formation, self-healing operations, assessment of reliability and the evaluation and learning loop for continuous reliability improvement.

Pseudo code

```
GPU-REL

Program to predict GPU failures and perform self-healing operations

    Initialize

        GPU Cluster

        Prediction Horizon H

        PCR Thresholds

        Self-Healing Policies

    Inputs

        ECC Errors

        GPU Temperature

        Power Consumption
```

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
        GPU Utilisation
        Memory Utilisation
        Fabric Errors
        Retransmissions
        Firmware Drift
        Storage Latency
        Topology Risk
        Maintenance History
        Prior Failures
    Models
        Random Forest
        XGBoost
        Random Survival Forest
    Start loop
        Collect GPU telemetry data
        Clean missing values
        Standardise numerical features
        Calculate ECC Trend
        Calculate Thermal Margin
        Calculate Failure Frequency
        Calculate Interconnect Instability
        Calculate Workload Intensity
        Calculate PCR Score
        Train Random Forest
        Train XGBoost
        Train Random Survival Forest
        Predict Failure Risk
        Estimate Time-to-Failure
        Apply SHAP Explainability
        Identify Important Risk Factors
        IF Failure Risk < Low Threshold
          THEN Continue Normal Operation
        IF Failure Risk >= Low Threshold
          THEN Monitor GPU
```

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
        IF Failure Risk >= Medium Threshold
          THEN Drain and Migrate Workload
        IF Failure Risk >= High Threshold
          THEN Execute Maintenance
        Validate GPU after Maintenance
        IF Validation = Pass
          THEN Reintegrate GPU
        IF Validation = Fail
          THEN Quarantine GPU
        Calculate Accuracy, Precision, Recall and F1
        Calculate ROC-AUC and MTTR
        Calculate Goodput and Operational Efficiency
        Store Outcomes as Feedback
        Update Predictive Models
      End loop
```

Fig. 8. Pseudo code

The pseudo code shows the operational logic of the framework when processing telemetry data, when generating predictive models to explain, PCR scoring, when taking actions to repair the network based on a threshold, when monitoring if actions are performed correctly or not and when continuing validation to improve reliability.

IV. RESULT AND DISCUSSION

The Results and Discussion segment assesses the productiveness, explain ability and self-healing architecture based on how well the model performs, the reliability results, and the impact on operational efficiency, for the various GPU-cluster configurations.

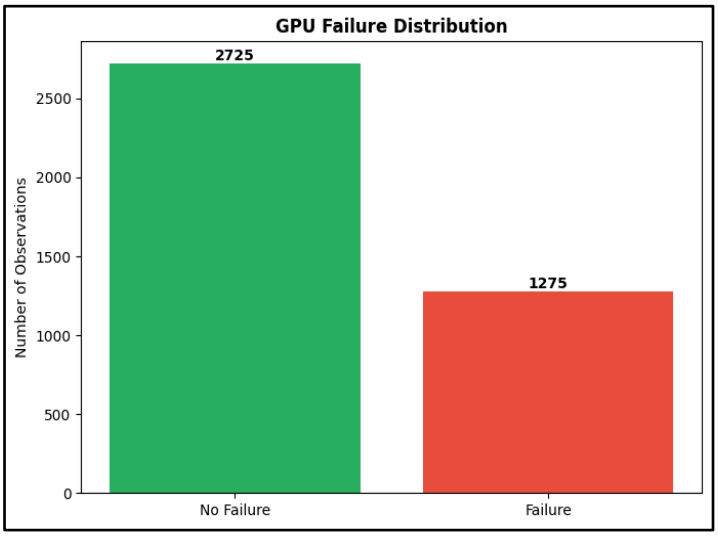

Fig. 9. GPU Failure Distribution

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The distribution has 2,725 non-failure perceptions and 1,275 failure perceptions, which is 68.1% and 31.9%. This moderate imbalance is consistent with static conditions of reliability in real life, while maintaining a robust failure/problem set for predictive learning tasks.

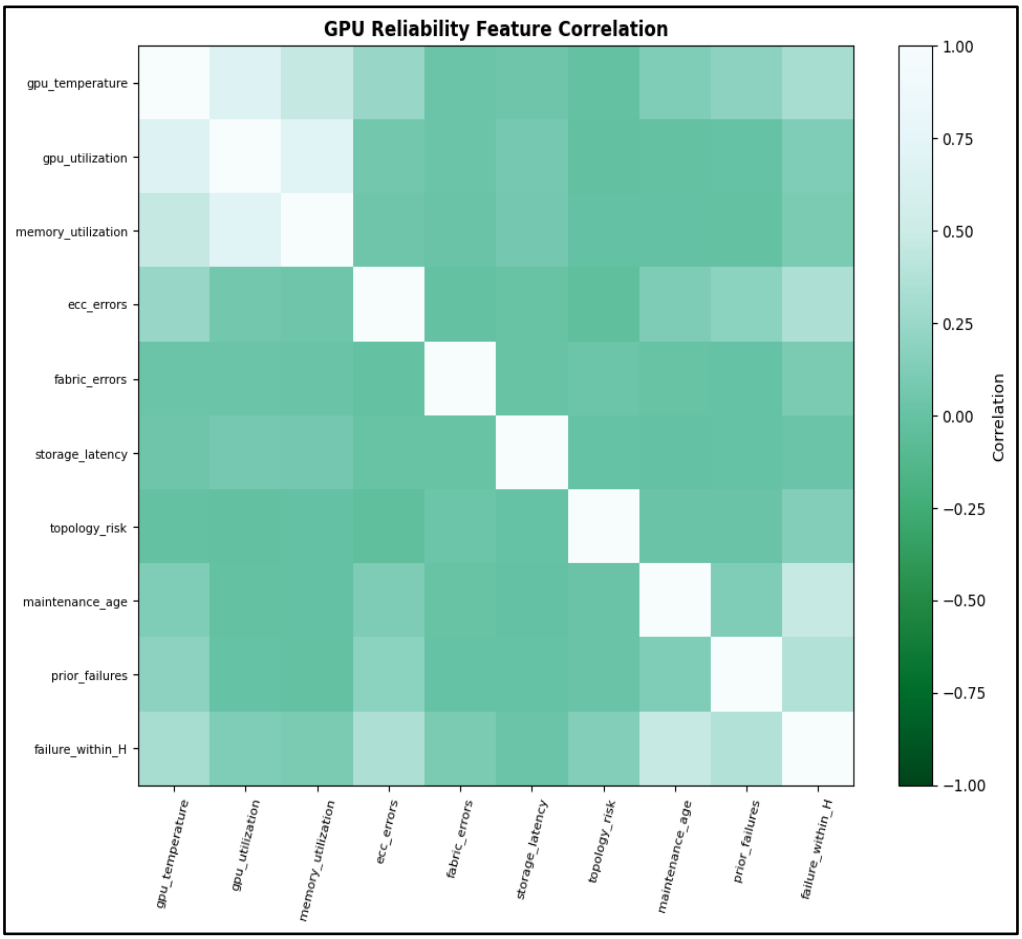

Fig. 10. GPU Reliability Feature Correlation

The correlation matrix shows generally weak to moderate correlations, and failure results are more clearly correlated with maintenance age, ECC errors, previous failures, topology risk and temperature, suggesting multivariate instead of single factor prediction.

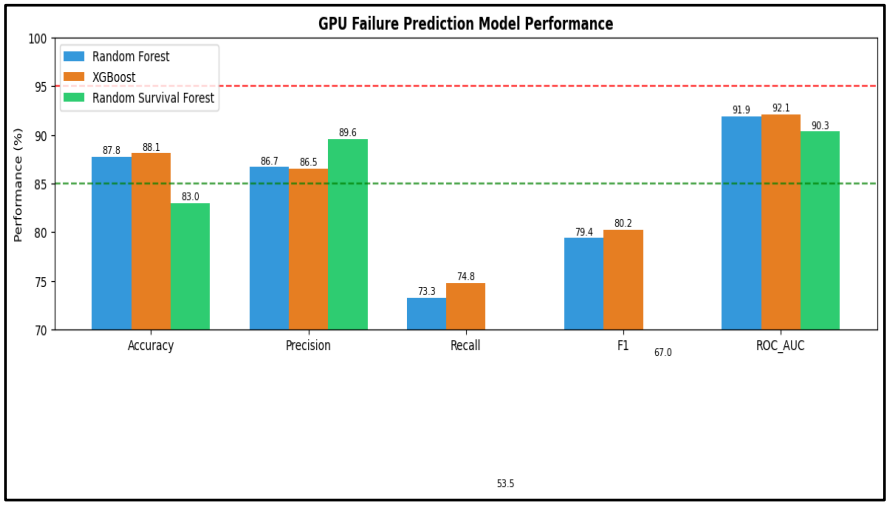

Fig. 11. GPU Failure Prediction Model Performance

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The comparison shows that overall XGBoost has more excellent performance, with its accuracy of 88.12% and ROC AUC score of 92.13%. Random Forest does not trail too far behind and has overall a highest precision of 88.52%, while Random Survival Forest is the highest in precision, at 89.61%, but its recall is poor. The performance chart indeed shows that XGBoost wins the competition in terms of accuracy (88.1%), recall (74.8%), F1 (80.2%) and ROC-AUC (92.1%). In general, there is a sensitivity specificity trade-off with accuracy over the various models with RSF performing best at 89.6%.

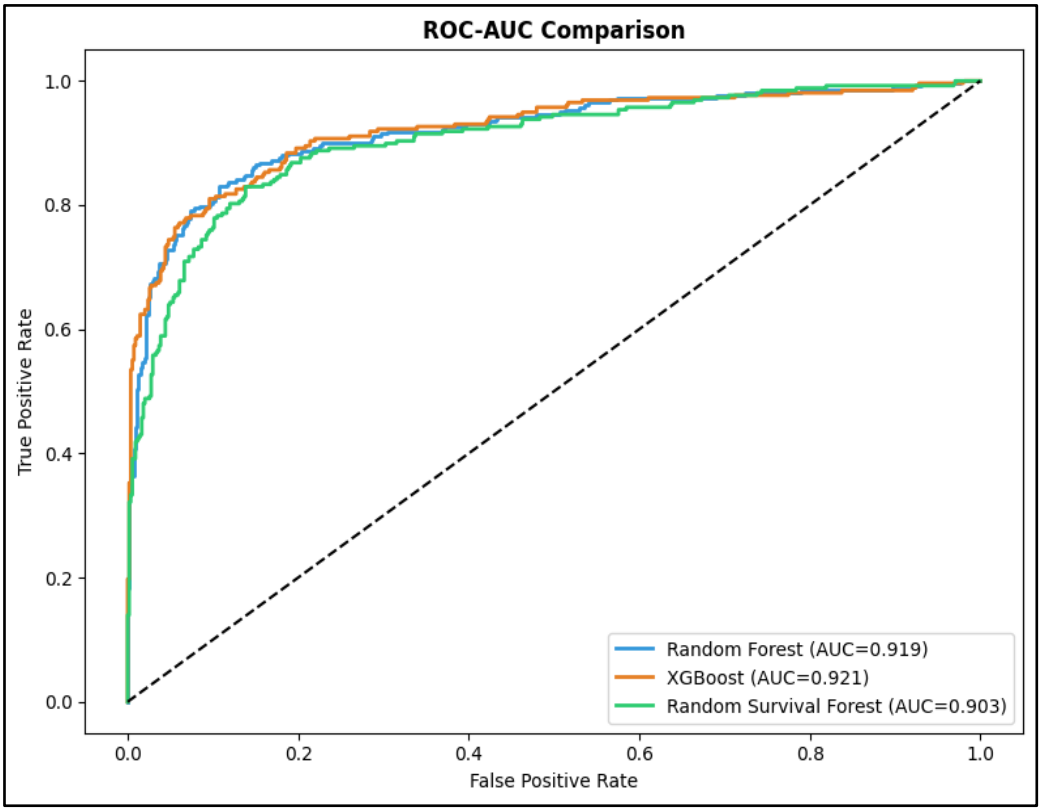

Fig. 12. ROC-AUC Comparison

All models show good discrimination, with XGBoost showing the highest AUC (0.921), Random Forest (0.919) and Random Survival Forest (0.903), which is significantly ahead of the random-classification diagonal.

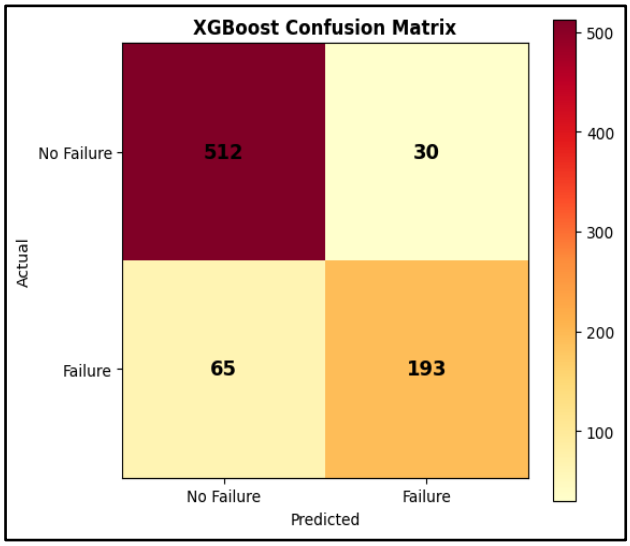

Fig. 13. XGBoost Confusion Matrix

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

XGBoost correctly predicted 512 normal cases (non-failures) and 193 failed cases with 30 false alarm cases and 65 false alarm cases missed. This pattern shows good specificity, although there are still some sensitivity concerns with failures.

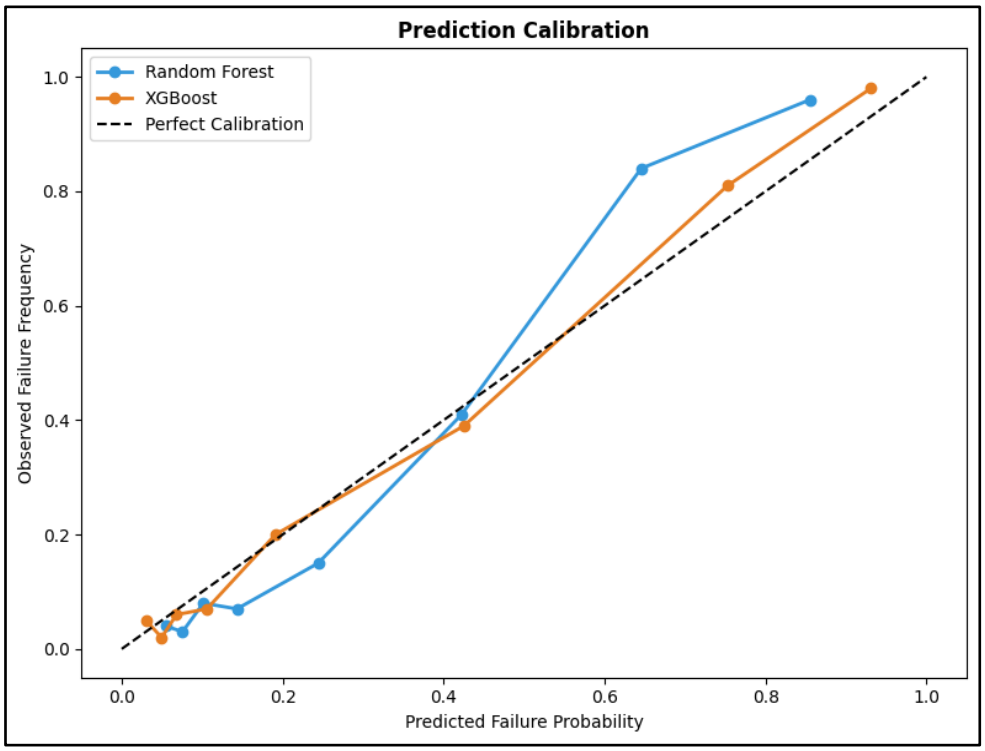

Fig. 14. Prediction Calibration

XGBoost predictions are still relatively close to perfect calibration over the various bins of probabilities while Random Forest has a tendency to overestimate the probability of failure in the range of probabilities of 0.65-0.85, raising doubts about its ability to make decisions on maintenance based on its predictions.

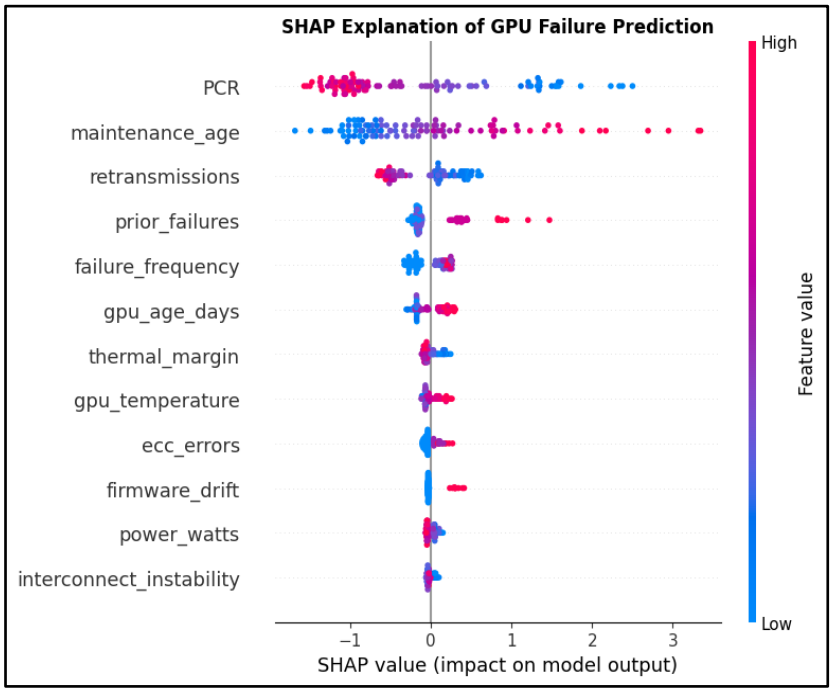

Fig. 15. SHAP Explanation of GPU Failure Prediction

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The main predictors considered by SHAP are maintenance age, retransmissions, failure number and failure frequency, in this order. Reduced PCR values represent risks for failures, and longer maintenance intervals result in positive contributions.

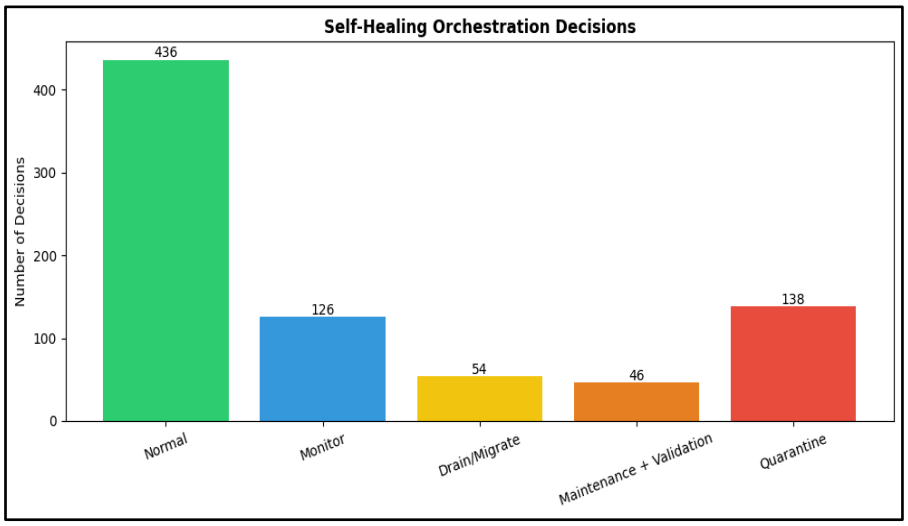

Fig. 16. Self-Healing Orchestration Decisions

The above bar chart reflects that the Normal decisions were predominant, accounting for 54.5%, which is 436 test observations. Quarantine accounted for 138 cases, Monitor 126, Drain/Migrate 54, and Maintenance + Validation 46 which mean that there is selective escalation rather than over-engaged autonomous disruption in infrastructure operations.

```
Interruptions Avoided: 147
False Predictive Drains: 37
Capacity Withheld: 29.75 %
Reintegration Success: 76.05 %
Autonomous Action Precision: 84.45 %
```

Fig. 17. Self-Healing Operational Performance Indicators

There are 37 false predictive drains and 147 times there are no interruptions with the self-healing framework. The level of capacity withholding was 29.75%, the percentage of those supported in reintegration was 76.05% and autonomous-action precision was 84.45%, suggesting significant reliability gains while maintaining a level of unnecessary intervention and loss of operational resources that is contained.

	Configuration	Interruptions	MTTR_Hours	Goodput_Percent
0	Reactive	258	5.5	94.62
1	Predictive	143	3.0	98.51
2	Self-Healing	111	1.5	99.42

Fig. 18. Operational Configuration Comparison

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The operational comparison shows that there were 258 interruptions identified during Reactive operation, while the Predictive control reduced the interruptions to 143 and with Self-Healing they went down to 111. Having improved the MTTR from 5.5 hours to 3.0 and 1.5 hours respectively, efficiency is demonstrated by the consequent increase in goodput from 94.62% to 98.51% and 99.42% respectively.

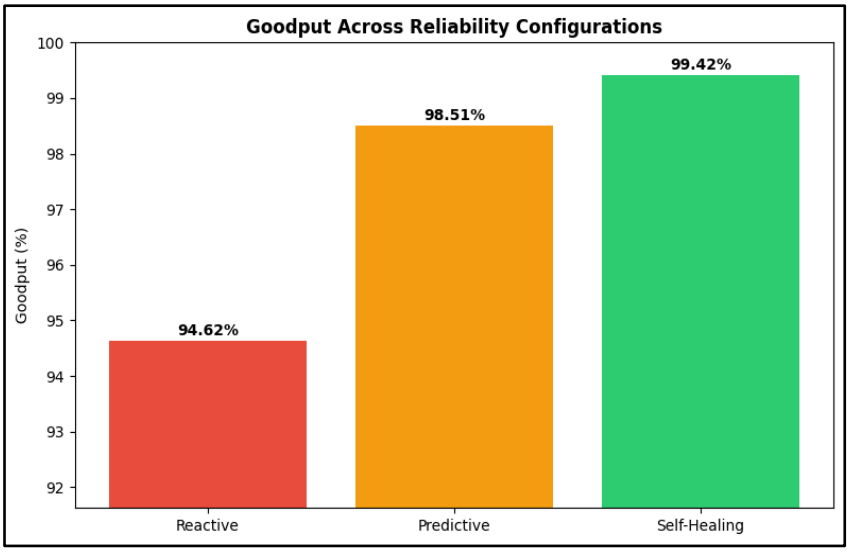

Fig. 19. Goodput across Reliability Configurations

The goodput graph shows that progressive, improvement was made in both Reactive and Predictive management and then with Self-Healing, the rate increased to 99.42%, from 94.62%.

TABLE II: SUMMARY OF PREDICTIVE RESULTS

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	ROC-AUC (%)
Random Forest	87.75	86.70	73.26	79.41	91.89
XGBoost	88.12	86.55	74.81	80.25	92.13
Random Survival Forest	83.00	89.61	53.49	66.99	90.34

DISCUSSION

The results are very promising with respect to the introduced approach of explainable self-healing reliability of hyper scale GPU clusters. XGBoost performed best overall in classification (88.12% accuracy, 92.13% ROC-AUC) and also showed good performance when compared with the other models at 87.75% accuracy. Random Survival Forest gave the lowest classification accuracy but its 0.8735 concordance index also indicated valuable time-to-failure ranking. In addition, SHAP analysis validated the transparency of the model, and emphasized the role played by the generators PCR, maintenance age, retransmissions, prior failures and thermal indicators. Self-healing resulted in major improvements of resilience, reducing interruptions by 258 (to 111) and MTTR by 5.5 (to 1,5) hours and goodput percent by 94.62% (to 99.42%).

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

LIMITATION

- Data reliability is a main issue that faced in this research where a synthetic data details are used. Its required a real world data that gives a butter understandings.
- The data imbalance is another issue in this research where some python libraries are used to handle this data imbalance issue in a proper way.

V. CONCLUSION

Conclusion

The results of this study show that explain ability, predictive, and self-healing reliability engineering can significantly boost hyper scale GPU operations. XGBoost topped the predictive performance, whereas Random Survival Forest added the most significant information about failure times. SHAP increased transparency and autonomous orchestration decreased interruptions, MTTR and wasted capacity.

Future Direction

Real-world hyper scale GPU telemetry and heterogeneous hardware generations, with production scale workloads, should be used in future research to test the proposed framework. More efforts are needed to explore online learning, concept drift adaptation, federated reliability modeling and RL-based maintenance policies [30]. Safe, scalable, and generalizable autonomous RMs across an ever-changing infrastructure of AI should also be explored through human-in-the-loop governance, uncertainty calibration, economic cost modeling, and cross-platform benchmarking.

VI. REFERENCES

[1] Polu, O.R., 2024. AI-driven prognostic failure analysis for autonomous resilience in cloud data centers. Journal ID, 2563, p.4512.
[2] Patel, J. and Shah, H., 2023. Machine learning and self-healing capabilities combined in adaptive AI architectures. International Research Journal of Engineering & Applied Sciences, 11(1), pp.10-55083.
[3] Prangon, N.F. and Wu, J., 2024. AI and computing horizons: cloud and edge in the modern era. Journal of Sensor and Actuator Networks, 13(4), p.44.
[4] Khan, A., Hassan, E.M., García, E. and Chen, D.L., 2021. Advancements in Software and Hardware Technologies: The Path to Smarter ICT Systems. International Journal of Information and Communication Technology Trends, 1(1), pp.118-126.
[5] Sannidhanam, A.H., 2023. Self-Healing Distributed Systems: AI-Driven Failure Prediction and Automated Recovery. International Journal of Research & Technology, 11(4), pp.168-174.
[6] Kaul, D., 2020. AI-driven fault detection and self-healing mechanisms in microservices architectures for distributed cloud environments. International Journal of Intelligent Automation and Computing, 3(7), pp.1-20.
[7] Rojas, E., Carrascal, D., Lopez-Pajares, D., Alvarez-Horcajo, J., Carral, J.A., Arco, J.M. and Martinez-Yelmo, I., 2024. A survey on ai-empowered softwarized industrial iot networks. Electronics, 13(10), p.1979.
[8] Canal, R., Hernandez, C., Tornero, R., Cilardo, A., Massari, G., Reghenzani, F., Fornaciari, W., Zapater, M., Atienza, D., Oleksiak, A. and PiĄtek, W., 2020. Predictive reliability and fault management in exascale systems: State of the art and perspectives. ACM Computing Surveys (CSUR), 53(5), pp.1-32.
[9] Sultana, M.S. and Akter, S., 2023. High-performance computing for scaling large-scale language and data models in enterprise applications. ASRC Procedia: Global Perspectives in Science and Scholarship, 3(1), pp.94-131.
[10] Sankar, T., Venkata Ramana, R.B. and Balamuralikrishnan, A., 2023. AI-Optimized Hyperscale Data Centers: Meeting the Rising Demands of Generative AI Workloads. International Journal of Trend in Scientific Research and Development, 7(1), pp.1504-1514.
[11] Bhatt, V., 2024. On continued significance of multimode links in data centers. Journal of Lightwave Technology, 43(4), pp.1700-1707.

[12] Bendechache, M., Svorobej, S., Takako Endo, P. and Lynn, T., 2020. Simulating resource management across the cloud-to-thing continuum: A survey and future directions. Future Internet, 12(6), p.95.
[13] Manganelli, M., Soldati, A., Martirano, L. and Ramakrishna, S., 2021. Strategies for improving the sustainability of data centers via energy mix, energy conservation, and circular energy. Sustainability, 13(11), p.6114.
[14] Khatun, Z. and Rahman, M.H., 2024. GPU-Accelerated Physics-Informed Digital Twins for Real-Time State Estimation and Fault Localization in Distribution Grids. American Journal of Scholarly Research and Innovation, 3(02), pp.179-216.
[15] Zhang, Y., Zhao, X., Li, Z., Cheng, G., Yin, J., Zhang, L. and Chen, Z., 2024. Integrating artificial intelligence into operating systems: A survey on techniques, applications, and future directions. arXiv preprint arXiv:2407.14567.
[16] Bharadwaj, M., 2022. Predictive Performance Modeling for Multi-Core and Many-Core Computing Architectures Using AI-Driven Analytics. American International Journal of Computer Science and Technology, 4(4), pp.1-13.
[17] Lavin, A., Gilligan-Lee, C.M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., Lange, D., Baydin, A.G., Sharma, A., Gibson, A. and Zheng, S., 2022. Technology readiness levels for machine learning systems. Nature Communications, 13(1), p.6039.
[18] Hasan, M.M. and Islam, M.M., 2022. High-performance computing architectures for training large-scale transformer models in cyber-resilient applications. ASRC Procedia: Global Perspectives in Science and Scholarship, 2(1), pp.193-226.
[19] Panyala, V.R., 2024. Designing self-healing cloud architectures for mission-critical distributed systems. International Journal of Science, Research and Technology, 7(2), pp.11717-11721.
[20] Vankayalapati, R.K., Pandugula, C., Ganti, V.K.A.T. and Mishra, G., 2022. AI-powered self-healing cloud infrastructures: A paradigm for autonomous fault recovery. Migration Letters, 19(6), pp.1173-1187.
[21] Alonso, J., Orue-Echevarria, L., Osaba, E., López Lobo, J., Martinez, I., Diaz de Arcaya, J. and Etxaniz, I., 2021. Optimization and prediction techniques for self-healing and self-learning applications in a trustworthy cloud continuum. Information, 12(8), p.308.
[22] Kotla, M.R.T., 2023. Autonomous enterprise integration: The future of self-healing data and API ecosystems. International Journal of Research and Applied Innovations, 6(3), pp.5968-5971.
[23] Johnphill, O., Sadiq, A.S., Al-Obeidat, F., Al-Khateeb, H., Taheir, M.A., Kaiwartya, O. and Ali, M., 2023. Self-healing in cyber–physical systems using machine learning: A critical analysis of theories and tools. Future Internet, 15(7), p.244.
[24] Liang, H. and Yin, X., 2023. Self-healing control: Review, framework, and prospect. IEEE Access, 11, pp.79495-79512.
[25] Ke, L., Gupta, U., Hempstead, M., Wu, C.J., Lee, H.H.S. and Zhang, X., 2022, April. Hercules: Heterogeneity-aware inference serving for at-scale personalized recommendation. In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 141-154). IEEE.
[26] Wei, D., Guo, C., Yang, L. and Xu, Y., 2023. A Loss Differentiation Method Based on Heterogeneous Ensemble Learning Model for Low Earth Orbit Satellite Networks. Entropy, 25(12), p.1642.
[27] Flores, V. and Leiva, C., 2021. A comparative study on supervised machine learning algorithms for copper recovery quality prediction in a leaching process. Sensors, 21(6), p.2119.
[28] Converso, G., Gallo, M., Murino, T. and Vespoli, S., 2023. Predicting failure probability in Industry 4.0 production systems: a workload-based prognostic model for maintenance planning. Applied Sciences, 13(3), p.1938.
[29] Angelopoulos, J. and Mourtzis, D., 2022. An intelligent product service system for adaptive maintenance of engineered-to-order manufacturing equipment assisted by augmented reality. Applied Sciences, 12(11), p.5349.
[30] Vallim Filho, A.R.D.A., Farina Moraes, D., Bhering de Aguiar Vallim, M.V., Santos da Silva, L. and da Silva, L.A., 2022. A machine learning modeling framework for predictive maintenance based on equipment load cycle: An application in a real world case. Energies, 15(10), p.3724.