Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion

Jiayi Nie[1], David Zheng[2]
[1] *Operations Research, Columbia University, NY, USA*
[2] *Information Technology, Carnegie Mellon University, PA, USA*
jiayinie8@gmail.com

DOI: 10.69987/JACS.2024.40409

Keywords

Cloud performance variability; noisy-neighbor risk; Azure VM Noise Dataset; anomaly detection; residual fusion; weak labels; placement replay; Server Machine Dataset.

Abstract

Cloud virtual machines can experience abrupt benchmark degradation when shared infrastructure is exposed to contention, maintenance activity, VM drift, storage-path effects, or unfavorable placement. This paper presents a noisy-neighbor-aware degradation-risk pipeline for benchmark traces without claiming host-level causal attribution. The study uses three public data sources: the official Azure VM Noise Dataset 2024, the Server Machine Dataset (SMD) from OmniAnomaly, and the Azure Trace for Packing 2020. The Azure experiment covers 7,037,220 raw measurements across 776 CSV partitions and evaluates a deterministic, per-partition balanced modeling table of 1,048,635 rows. Because Azure does not publish co-tenant identities or root-cause labels, Azure F1 scores are reported against weak observed-degradation labels derived from severe raw performance and runtime tails rather than injected synthetic labels. SMD provides an external label-based sanity check with 708,420 test points and official anomaly labels. The proposed score is renamed NN-Aware Residual Fusion (NN-RF): it combines signed performance residuals, runtime residuals, temporal deltas, and previous-day cell pressure through a non-negative validation-calibrated fusion model; PCA reconstruction is treated as a candidate feature rather than as a temporal autoencoder. On the held-out Azure period, NN-RF achieves F1 0.938, AUROC 0.998, and recall 0.932 under the weak degradation task, while Robust MAD remains a strong residual baseline with F1 0.932. On SMD, a windowed neural autoencoder gives the best held-out F1, 0.299, among the evaluated external baselines. A capacity-constrained replay of 50,000 Azure Packing Trace admissions shows that heat-aware first-fit keeps the same acceptance rate as trace-order first-fit while reducing mean placement risk by 7.0%. The results support a narrower conclusion: benchmark residuals can provide useful risk signals for placement, but they should not be interpreted as definitive noisy-neighbor root-cause labels.

Introduction

Cloud resource management is not only a capacity-allocation problem. A VM can receive the requested vCPUs, memory, and disks while still observing degraded benchmark behavior because contention, background services, maintenance, storage paths, and placement choices affect hidden shared resources. The operational concern is tail behavior: a small number of degraded tasks can dominate end-to-end service latency [1]. Large-scale schedulers and cluster managers therefore treat placement, heterogeneity, and interference as first-order

concerns rather than post-deployment tuning details [2]-[5].

The phrase noisy neighbor is useful as an operational shorthand, but public benchmark traces rarely contain the host-level evidence needed to prove that one tenant caused another tenant's slowdown. This study therefore separates risk modeling from causal attribution. The Azure VM Noise Dataset is used to model observed benchmark degradation risk; SMD is used to check whether the same anomaly-detection pipeline behaves sensibly on a labeled server-machine anomaly benchmark; and the Azure Packing Trace is used to replay placement decisions under capacity constraints.

The method claim is deliberately narrow. NN-RF is not a temporal autoencoder; it is a residual-fusion detector with an optional reconstruction feature. The method uses non-negative validation-calibrated weights instead of fixed hand-tuned weights, and it uses previous-day peer pressure rather than contemporaneous peer aggregation to reduce leakage risk.

The paper makes four contributions. First, it defines a unit-aware normalization procedure for heterogeneous Azure benchmark metrics. Second, it evaluates NN-Aware Residual Fusion against robust, classical, and neural anomaly baselines under a strict temporal split. Third, it adds external validation on SMD with official labels and modern neural baselines, including DeepSVDD and a windowed autoencoder. Fourth, it converts Azure risk scores into heat features and tests them in a capacity-constrained replay using the Azure Packing Trace.

Method

Figure 1 summarizes the end-to-end pipeline. Raw Azure benchmark partitions are first normalized within each measurement unit. A measurement unit is defined by benchmark suite, test name, unit, VM lifespan, region, and SKU, which prevents unrelated metrics such as disk latency, Redis throughput, and PostgreSQL latency from being compared on the same raw scale. The normalized table is then scored by baselines and by NN-RF. Validation data are used only for threshold and fusion calibration, and the final test period is evaluated once.

The official Azure raw directory contains 776 CSV files with value, runtime, starttime, and VM_id columns. Sixteen files are empty in this snapshot, leaving 760 nonempty partitions. The raw directory contains 7,037,220 rows from 2023-05-28 to 2024-09-23. To avoid letting very long partitions dominate the experiment, the Azure modeling table is a deterministic balanced sample capped at 1,500 rows per partition, giving 1,048,635 rows while preserving all nonempty benchmark families and the full time range.

For each measurement unit, the training period supplies the median and median absolute deviation of value and runtime. Direction metadata converts raw values into signed performance residuals, z_perf, where positive values always mean worse performance. Latency-like metrics use value minus median; throughput-like metrics use median minus value. Runtime is normalized separately as z_runtime. The pipeline also computes one-step lag residuals, positive temporal deltas, hour and day-of-week encodings, and previous-day peer pressure. Peer pressure is the previous day's median positive residual for the same region, SKU, lifespan, and resource class. It intentionally excludes same-day measurements to avoid encoding the label presence of contemporaneous disturbances.

Because Azure does not expose co-tenant identities, physical host IDs, maintenance events, or root-cause labels, Azure F1 and confusion matrices use weak observed-degradation labels. A row is labeled as degraded when it shows a severe signed performance residual, or a moderately severe residual accompanied by elevated runtime or a worsening temporal delta. These labels are derived only from observed benchmark values and runtime; they are not injected anomalies and are not presented as proof of noisy-neighbor causality. The external SMD experiment uses official anomaly labels and is reported separately.

NN-RF uses five candidate terms: PCA reconstruction error, positive signed residual, positive runtime residual, previous-day peer pressure, and positive temporal delta. Instead of fixed hand-written weights, the final fusion weights are obtained by a non-negative logistic calibration on the Azure validation period. This makes the score

auditable: if reconstruction does not add validation value after residual terms are present, its coefficient can be zero. The calibrated score is used for the Azure test period and for cell-level heat maps.

The Azure split is temporal. Rows before 2024-03-01 form the fitting pool, rows from 2024-03-01 through 2024-06-14 form validation, and rows from 2024-06-15 onward form the held-out test period. For SMD, each machine's official test sequence is split chronologically into validation and held-out halves after fitting on the official training sequence. For the packing replay, the experiment uses the first 50,000 admissions after time zero from the Azure Packing Trace and compares trace-order first-fit, heat-aware first-fit, and priority heat-aware placement under normalized CPU, memory, SSD, and NIC constraints.

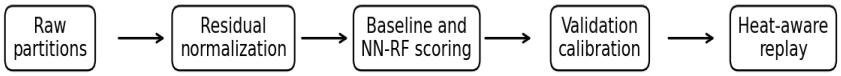

Fig. 1. End-to-end data preparation, residual-fusion scoring, external validation, and placement-replay pipeline.

Table 1. Data sources and evaluation roles.

Data source	Role	Scale	Label use	Split or replay
Azure VM Noise Dataset 2024	Main benchmark degradation-risk task	776 CSV files; 7,037,220 raw rows; 1,048,635 balanced modeling rows	Weak observed-degradation labels	Temporal train/validation/test
Server Machine Dataset (SMD)	External labeled anomaly validation	28 machines; 708,405 train rows; 708,420 test rows	Official anomaly labels	First test half for validation; second half held out
Azure Trace for Packing 2020	Capacity-constrained placement replay	5,559,800 VM rows; 4,619 VM-type rows	Uses Azure heat score as risk prior	First 50,000 admissions after time zero

Table 2. Azure VM Noise raw benchmark-suite coverage.

Benchmark Suite	Files	Nonempty_Files	Rows	Units	Test_Names
Flexible IO Tester	352	352	2,665,669	3	16

Benchmark Suite	Files	Nonempty_Files	Rows	Units	Test_Names
Intel Memory Latency Checker	88	88	1,024,330	2	11
OSBench	40	40	465,786	2	5
PostgreSQL	176	160	1,489,674	2	32
Redis	40	40	460,618	1	5
Stress-NG	32	32	372,431	1	4
Sysbench	16	16	186,288	2	2
perf-bench	32	32	372,424	2	4

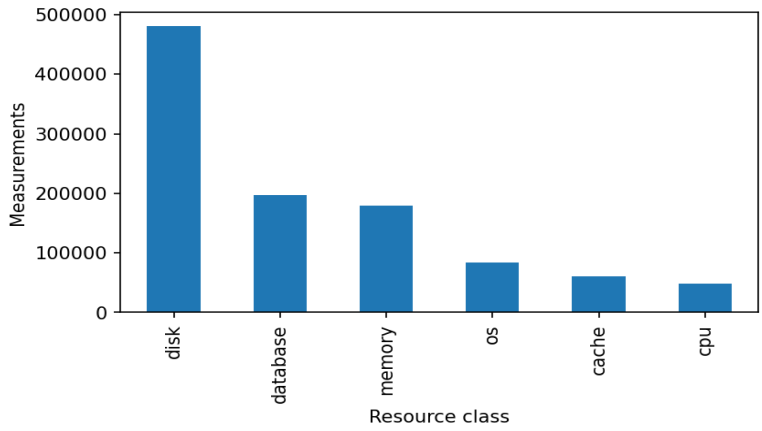

Fig. 2. Balanced Azure modeling-table coverage by resource class.

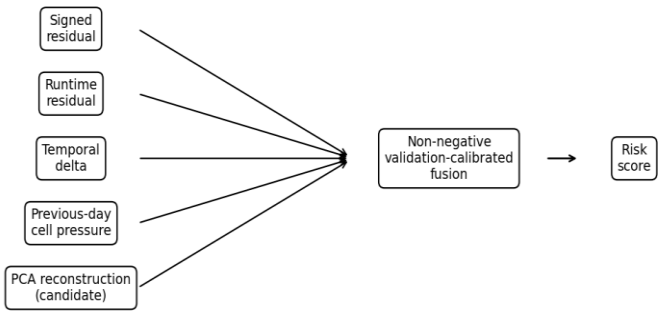

Fig. 3. NN-RF feature path and validation-calibrated fusion score.

Results and Discussion

Table 3 reports the Azure held-out results under the weak observed-degradation task. NN-RF obtains the best F1, 0.938, and AUROC, 0.998. Robust MAD is close behind with F1 0.932 and higher recall, which is expected because the Azure labels are based on severe residual and runtime degradation. The result should therefore be read as a degradation-risk detection result rather than as causal noisy-neighbor attribution.

Figure 4 shows the ROC curves, and Figure 5 shows the F1 comparison. The interpretation is intentionally conservative: the robust residual baseline is the strongest simple alarm, while NN-RF is a calibrated risk score that is useful when row-level residuals, runtime, temporal changes, and cell history must be combined in one ranking signal.

Table 3. Azure held-out model comparison under weak observed-degradation labels.

Model	F1	Precision	Recall	AUROC	AUPRC	Threshold
NN-Aware Residual Fusion	0.938	0.943	0.932	0.998	0.990	2.060
NN-RF without peer lag	0.937	0.944	0.930	0.998	0.990	2.098
Robust MAD	0.932	0.881	0.990	0.997	0.984	3.000
PCA Reconstruction	0.551	0.439	0.738	0.826	0.582	0.143
MLP Autoencoder	0.551	0.568	0.534	0.818	0.653	0.174
One-Class SVM	0.424	0.841	0.284	0.837	0.611	-3.123
EWMA Residual	0.383	0.589	0.283	0.667	0.397	3.003
Isolation Forest	0.374	0.280	0.565	0.724	0.307	0.528

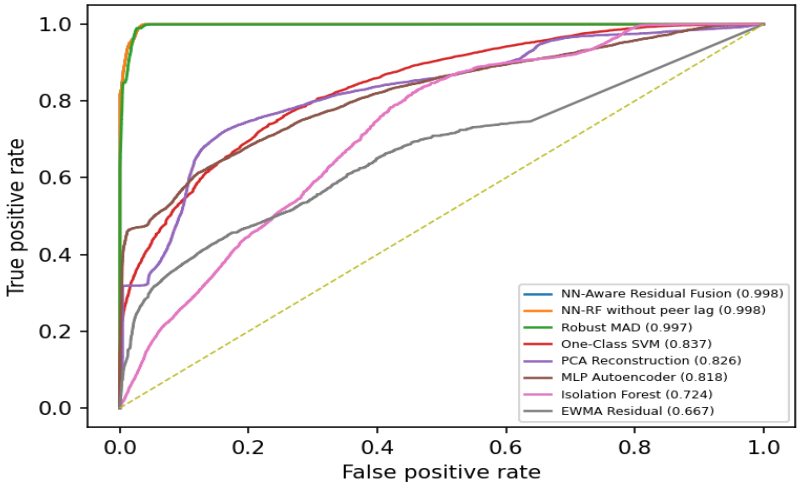

Fig. 4. ROC curves for Azure held-out weak observed-degradation detection.

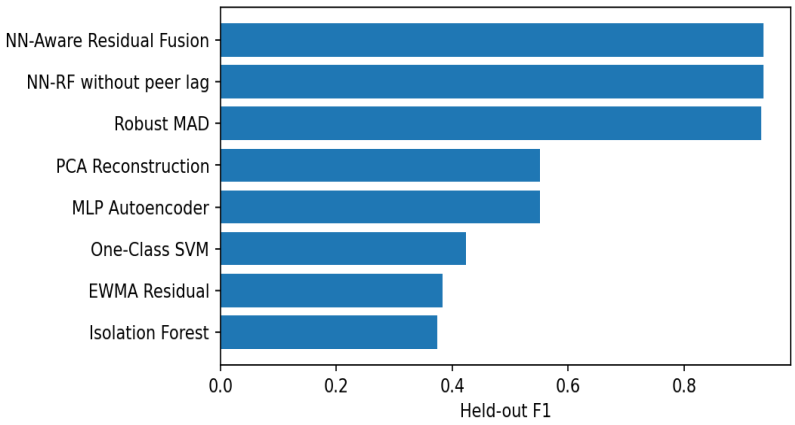

Fig. 5. Azure held-out F1 comparison for robust, classical, reconstruction, and fusion detectors.

Table 4 breaks down NN-RF by resource class. Disk has the strongest F1, followed by memory and operating-system measurements. Cache and CPU have lower sample counts in the held-out period and lower F1, but their AUROC remains high. These differences support reporting resource-level metrics instead of relying only on aggregate performance.

Table 4. Azure NN-RF held-out performance by resource class.

Resource	Samples	Event_Rate	F1	Precision	Recall	AUROC	AUPRC
cache	3,452	0.065	0.778	1.000	0.637	1.000	0.996
cpu	2,738	0.043	0.836	0.851	0.822	0.996	0.917
database	11,212	0.148	0.844	0.835	0.853	0.990	0.949
disk	23,179	0.208	0.982	0.976	0.989	1.000	0.999

Resource	Samples	Event_Rate	F1	Precision	Recall	AUROC	AUPRC
memory	10,352	0.169	0.937	0.988	0.891	0.999	0.996
os	4,758	0.128	0.908	0.879	0.939	0.996	0.979

Table 5 gives the calibrated fusion weights. The positive signed residual dominates the Azure weak-label task, while runtime and temporal-delta terms add smaller contributions. The PCA reconstruction term is retained as a candidate feature, but the non-negative calibration assigns it zero weight after the residual terms are present. This is the main reason the revised method is not described as a temporal autoencoder.

Table 5. Non-negative validation-calibrated NN-RF coefficients.

Feature	Coefficient
PCA reconstruction	0.000
positive signed residual	5.031
positive runtime residual	0.214
previous-day peer pressure	0.068
positive temporal delta	0.342

Table 6 addresses the peer-pressure concern. The full NN-RF score and the no-peer variant are nearly tied; the full model has slightly higher F1 and AUROC, while the no-peer variant has slightly higher precision. Because the peer feature uses only previous-day cell pressure, it avoids the same-day leakage risk of contemporaneous aggregation.

Table 6. Azure ablation and residual-component comparison.

Model	F1	Precision	Recall	AUROC	AUPRC
NN-Aware Residual Fusion	0.938	0.943	0.932	0.998	0.990
NN-RF without peer lag	0.937	0.944	0.930	0.998	0.990
PCA Reconstruction	0.551	0.439	0.738	0.826	0.582
Robust MAD	0.932	0.881	0.990	0.997	0.984
EWMA Residual	0.383	0.589	0.283	0.667	0.397

Table 7 gives the held-out confusion matrix for the NN-RF operating point selected on validation data. The matrix is shown as counts rather than only rates so that the operating point can be audited directly.

Table 7. Azure held-out confusion matrix for NN-RF.

Actual class	Predicted normal	Predicted degraded
True normal	45,994	513
True degraded	627	8,557

Table 8 reports threshold sensitivity. Conservative placement can tolerate lower thresholds and higher recall, whereas incident escalation would typically select a higher threshold for precision.

Table 8. Azure NN-RF threshold sensitivity.

Validation score percentile	F1	Precision	Recall
0.800	0.755	0.607	1.000
0.850	0.876	0.780	1.000
0.900	0.901	0.986	0.829
0.925	0.747	1.000	0.596
0.950	0.638	1.000	0.468
0.975	0.240	1.000	0.136
0.990	0.061	1.000	0.031

Table 9 summarizes resource-level variability in the balanced Azure modeling table. Disk and memory show the largest positive residual tails, which explains why these resources dominate heat-map behavior. Figure 6 visualizes average held-out risk by cell and resource, while Figure 7 shows a temporal slice of risk scores and weak labels.

Table 9. Azure resource-level variability in the balanced modeling table.

Resource	Rows	Weak_Event_Rate	Median_Positive_Residual	P95_Positive_Residual	Median_Runtime
cache	60,000	0.084	0.000	3.125	44.170
cpu	48,000	0.077	0.000	4.338	30.040
database	196,444	0.104	0.000	5.472	337.990
disk	480,191	0.116	0.000	20.000	85.330

Resource	Rows	Weak_Event_Rate	Median_Positive_Residual	P95_Positive_Residual	Median_Runtime
memory	180,000	0.133	0.000	8.433	28.020
os	84,000	0.112	0.055	6.110	5.060

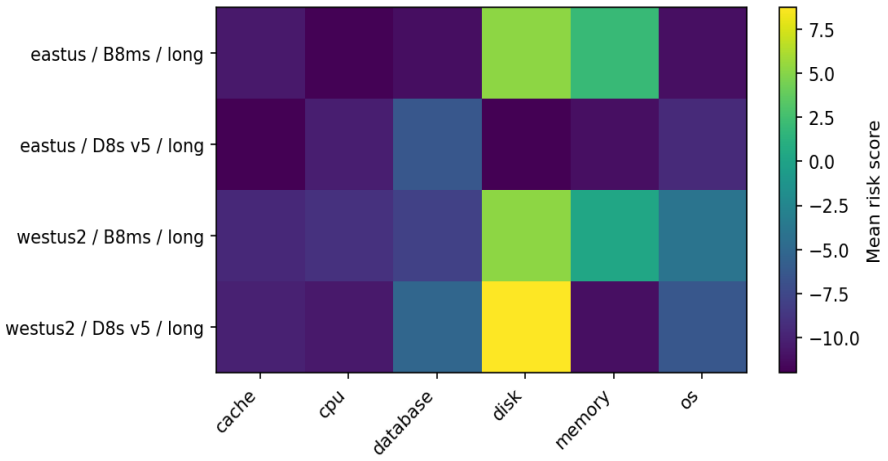

Fig. 6. Azure held-out heat map by region, SKU, lifespan, and resource class.

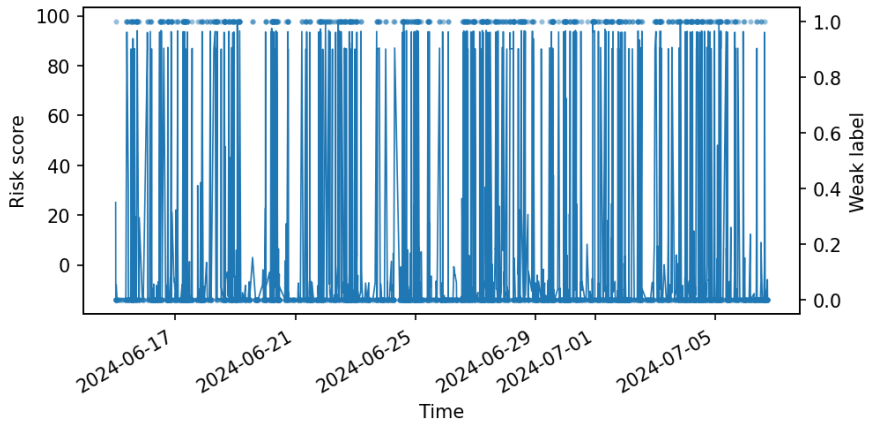

Fig. 7. Temporal slice of Azure NN-RF scores and weak observed-degradation labels.

The SMD experiment checks whether the evaluation pipeline behaves sensibly when official anomaly labels are available. Table 10 shows that the windowed autoencoder has the highest held-out F1, 0.299, followed by PCA reconstruction and DeepSVDD. The absolute F1 values are lower than the Azure weak-label scores because SMD labels cover heterogeneous operational anomalies rather than a residual-defined degradation rule. This external result is useful precisely because it prevents the paper from relying only on residual-derived Azure labels.

Figure 8 visualizes the SMD F1 results. The finding is not that NN-RF is a universal anomaly model; rather, the evidence shows that residual fusion is strong for Azure benchmark degradation, while windowed neural reconstruction is more competitive on a labeled multivariate server-machine anomaly benchmark.

Table 10. External SMD held-out anomaly-detection validation.

Model	F1	Precision	Recall	AUROC	AUPRC	Threshold
Windowed Autoencoder	0.299	0.289	0.309	0.770	0.226	0.484
PCA Reconstruction	0.265	0.297	0.240	0.749	0.201	0.760
DeepSVDD	0.251	0.161	0.564	0.735	0.199	0.001
MLP Autoencoder	0.247	0.151	0.677	0.775	0.222	0.073
Isolation Forest	0.216	0.144	0.434	0.665	0.229	0.460
EWMA Residual	0.214	0.264	0.181	0.764	0.209	0.366
Robust MAD	0.141	0.231	0.101	0.569	0.117	9,639,366

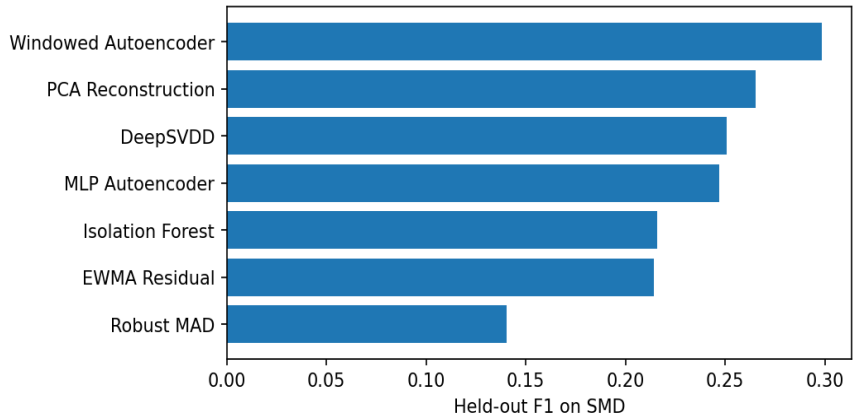

Fig. 8. Held-out F1 on the external SMD labeled anomaly task.

The placement experiment uses a capacity-constrained replay rather than a hottest-quartile filter. Table 11 reports the first 50,000 chronological VM admissions after time zero from the Azure Packing Trace. Heat-aware first-fit keeps the same acceptance rate as trace-order first-fit and reduces mean placement risk by 6.96%. Priority heat-aware placement reserves lower-risk cells for priority-zero

VMs; it lowers mean risk by 4.13% but accepts slightly fewer VMs in this simple replay.

Figure 9 shows the relative risk reductions. The replay is still not a production scheduler: it does not model migration, fairness over long horizons, locality constraints, or tenant-specific affinity. It is nevertheless stronger than a pure hottest-quartile demonstration because it uses VM resource demands and capacity constraints from an independent packing trace.

Table 11. Azure Packing Trace capacity-constrained placement replay.

Policy	Accepted VMs	Rejected VMs	Acceptance_ Rate	Mean_Placement_Risk	P0_Mean_Risk	Relative_Risk_Reduction
Trace-order first-fit	19,384	30,616	0.388	0.109	0.109	0.000
Heat-aware first-fit	19,384	30,616	0.388	0.102	0.102	0.070
Priority heat-aware	19,256	30,744	0.385	0.105	0.102	0.041

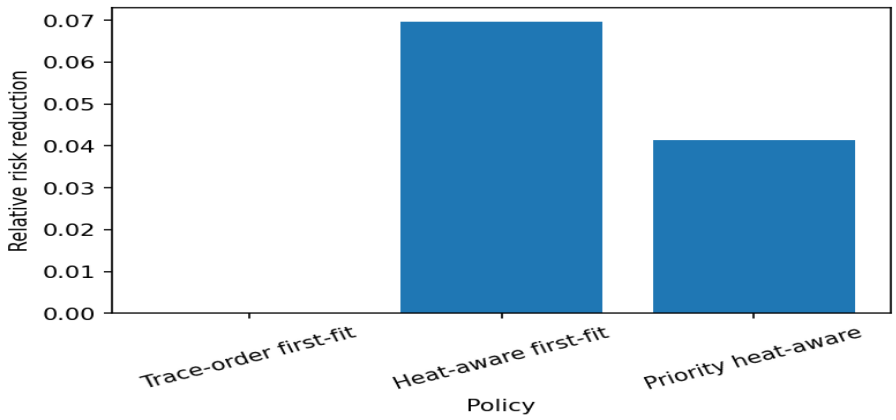

Fig. 9. Relative placement-risk reduction in the Azure Packing Trace replay.

Limitations

The main limitation is causality. A high Azure risk score identifies a benchmark measurement or placement cell that resembles observed degradation, but it does not prove that a co-located tenant caused the slowdown. Host counters, co-tenant identities, maintenance records, and controlled co-location experiments would be needed for root-cause attribution.

The second limitation is label realism. Azure labels in this paper are weak observed-degradation labels based on raw residual and runtime tails. They are more realistic than injected synthetic labels because they come directly from the official benchmark measurements, but they are still not ground-truth noisy-neighbor labels. The SMD experiment partly mitigates this issue by adding an external official-label benchmark, but SMD is a general server-machine anomaly dataset rather than a cloud co-location dataset.

The third limitation is feature observability. The Azure VM Noise benchmark set covers cache, CPU, disk, memory, OS behavior, PostgreSQL, and Redis, but it does not include network benchmarks. The model therefore cannot detect network-path

interference directly. Finally, the packing replay is a simplified scheduler study; production placement would need additional constraints for fairness, migration cost, affinity, anti-affinity, and capacity fragmentation.

Conclusion

This paper frames the study as VM benchmark degradation-risk modeling rather than as a causal noisy-neighbor attribution system. NN-RF is an auditable residual-fusion score calibrated on validation data. It uses previous-day peer pressure to reduce leakage risk and reports ablations so that row-level detection and placement heat scoring can be interpreted separately.

On Azure, NN-RF and Robust MAD are both strong under weak observed-degradation labels, with NN-RF reaching F1 0.938 and AUROC 0.998 on the held-out period. On SMD, the windowed autoencoder is strongest among the external baselines, showing that residual-fusion performance on Azure should not be overgeneralized. In the Azure Packing Trace replay, heat-aware first-fit reduces mean placement risk without reducing acceptance rate. Together, these results support a modest operational conclusion: normalized benchmark residuals can be useful placement-risk signals, but causal noisy-neighbor diagnosis requires additional host-level evidence.

References

[1] J. Dean and L. A. Barroso, "The tail at scale," Communications of the ACM, vol. 56, no. 2, pp. 74-80, 2013.

[2] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in Proc. EuroSys, 2015, pp. 1-17.

[3] C. Delimitrou and C. Kozyrakis, "Paragon: QoS-aware scheduling for heterogeneous datacenters," in Proc. ASPLOS, 2013, pp. 77-88.

[4] C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and QoS-aware cluster management," in Proc. ASPLOS, 2014, pp. 127-144.

[5] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega: Flexible, scalable schedulers for large compute clusters," in Proc. EuroSys, 2013, pp. 351-364.

[6] Microsoft Azure, "Azure VM Noise Dataset 2024," Azure Public Dataset repository, 2024.

[7] Microsoft Azure, "Azure Trace for Packing 2020," Azure Public Dataset repository, 2020.

[8] Chenyu Li, Wenhao Su, and Eric Zhang, “Lightweight Hallucination Firewall for Enterprise LLM Applications: Evidence Consistency, Self-Checking, and Small-Model Detection on TruthfulQA”, JACS, vol. 3, no. 1, pp. 49–65, Jan. 2023, doi: 10.69987/JACS.2023.30104.

[9] Jiaying Jin, Tina Huang, and Sam Lu, “Cost-Sensitive Learning, Simulated PU Learning, and One-Class Autoencoding for Extreme-Imbalance Credit Card Fraud Detection”, JACS, vol. 4, no. 6, pp. 64–73, Jun. 2024, doi: 10.69987/JACS.2024.40605.

[10] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in Proc. IEEE ICDM, 2008, pp. 413-422.

[11] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol. 13, no. 7, pp. 1443-1471, 2001.

[12] L. Ruff et al., "Deep one-class classification," in Proc. ICML, 2018, pp. 4393-4402.

[13] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in Proc. KDD, 2018, pp. 387-395.

[14] Shenghan Lu and David Zhou, “TinyLLM-Assisted Intrusion Detection for Real-Time IoT Networks”, JACS, vol. 4, no. 8, pp. 72–87, Aug. 2024, doi: 10.69987/JACS.2024.40809.