Outlier and Anomaly-Handling for 6G Wireless Measurement Data

A Systematic, Downstream-Centric Comparison of Statistical Filters and Unsupervised Outlier Detectors for Tabular and Time-Series 6G Network Measurements

Matei-Alexandru Stanescu [1]
Supervisor(s):Rihan Hai[1]**, Yuandou Wang**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 20, 2026

Name of the student: Matei-Alexandru Stanescu (M.A.Stanescu@student.tudelft.nl)
Final project course: CSE3000 Research Project
Thesis committee: Rihan Hai, Yuandou Wang, Julian Urbano Merino

Abstract

Machine-learning-driven management of next-generation (6G) networks depends on measurements that are routinely corrupted by sensor noise, hardware imperfections, bursty interference, and malicious activity, so outlier handling is widely assumed to be a prerequisite for reliable downstream models. Whether, and which, cleaning methods actually help, and whether this differs across data modalities, remains unclear. Using two *real, labelled* datasets with no synthetic contamination, attack traffic from a functional 5G testbed (attack classification) and an operational web-latency KPI series (short-term forecasting), we systematically compare six outlier-handling methods, namely interpretable statistical filters (robust Z-score replacement, IQR clipping, and Savitzky–Golay smoothing) and unsupervised detectors (Isolation Forest, Local Outlier Factor, and PCA reconstruction), against a no-cleaning baseline. Hyperparameters are tuned without access to held-out labels; each method is evaluated under both a robust (Random Forest) and a noise-sensitive (k-NN) downstream model, with paired significance tests and false-discovery-rate (FDR) correction, a detection diagnostic, and runtime. The result is largely negative: after FDR correction, no method significantly improves downstream performance on either modality. Savitzky–Golay smoothing gives the only suggestive forecasting gain ($\approx$17% lower error under Random Forest) but does not survive correction; deletion- and clipping-based methods are neutral-to-harmful (IQR significantly *degrades* classification); and the unsupervised detectors rank real attacks barely above chance (ROC-AUC 0.54–0.60), even though a supervised model separates the same classes at 0.86, statistical outlier detection is a poor proxy for the anomalies of interest. As the slowest detectors are also the most harmful and exceed the near-real-time control budget, we conclude that a generic outlier-handling stage offers no reliable benefit for these tasks: its value must be demonstrated rather than assumed, with lightweight smoothing the only candidate worth trying on noisy sequential signals.

1 Introduction

The emergence of sixth-generation (6G) wireless networks promises an unprecedented convergence of heterogeneous devices, intelligent radio access nodes, and experimental testbeds operating at a scale far exceeding any previous generation. These systems continuously generate high-dimensional measurements: signal-to-noise ratios, throughput figures, latency traces, and packet-level traffic statistics, that form the raw material for the machine learning models expected to drive autonomous network management [18, 4]. Yet this data arrives compromised: sensor noise, hardware imperfections, bursty interference, and malicious activity corrupt even routine KPI (Key Performance Indicators) logs, and the presence of outliers in such measurements is not a peripheral concern but a central one. As Aguinis et al. [2] demonstrate across a comprehensive taxonomy of 14 outlier definitions, 39 identification techniques, and 20 handling techniques, the way outliers are defined, identified, and handled directly alters conclusions: a finding that translates with full force to any data-driven system, including ML-driven 6G network management. Unreliable input data directly degrades the accuracy of downstream tasks such as handover prediction, resource allocation, and anomaly-based fault detection [19].

Early responses to this problem relied on rule-based thresholds and simple statistical filters, but such approaches proved too rigid for the dynamic, high-dimensional nature of wireless environments [7]. More recent work has turned to ML-based methods: Isolation Forest and Local Outlier Factor have been applied to tabular KPI logs from O-RAN deployments [3], while autoencoder-based architectures, including variational autoencoders combined with recurrent transformers, have demonstrated strong performance on sequential network traffic [21]. Wavelet-assisted Bayesian deep learning has been explored for detecting malicious data injection at 6G edges, showing that frequency-domain denoising can complement learned outlier scoring [23].

The breadth of available approaches is substantial; Boukerche et al. [5] survey methods across proximity-based, projection-based, deep learning-based, and semi-supervised paradigms, covering nearest-neighbor techniques such as LOF [6], tree-based ensemble methods such as Isolation Forest [15], and deep approaches such as robust autoencoders [26], highlighting both the richness of the field and the difficulty of selecting the right method for a given task. Despite this progress, a systematic comparative evaluation of statistical versus ML-based denoising and outlier-handling methods, specifically assessing their downstream impact on ML model performance across both tabular and time-series modalities of real 6G network measurements, remains largely absent from the literature. This paper addresses this gap. The central research question is

How do different denoising and outlier-handling methods compare in preserving data utility for machine learning tasks when applied to noisy 6G network measurements across tabular and time-series modalities, and, based on this comparison, which strategy is most suitable for the dominant noise and anomaly types observed in each modality?

The study answers this using *real, labelled* 6G-relevant measurement data rather than synthetically corrupted benchmarks, comparing six outlier-handling methods, three interpretable statistical filters (Z-score filtering, IQR clipping, and Savitzky–Golay smoothing, the last applied to the time-series modality only) and three unsupervised detectors (Isolation Forest, Local Outlier Factor, and **PCA-based reconstruction, a lightweight linear autoencoder**) , against a no-cleaning baseline, with all cleaning hyperparameters selected by non-oracle tuning that uses no clean ground truth. We decompose the central research question into three subquestions:

SQ1: Does any outlier-handling method improve *downstream* model performance over no cleaning, and if so, which is most effective, on each modality: network attack classification on labelled 5G-testbed traffic (macro-F1, ROC-AUC) and one-step latency forecasting on operational service KPIs (RMSE, MAE)?

SQ2: Is any such benefit *conditional*: does the most effective strategy differ between the tabular and time-series modalities, and does it depend on whether the downstream model is robust (Random Forest) or noise-sensitive (k-nearest neighbours)?

SQ3: Do the unsupervised detectors actually *identify* the real, labeled anomalies they are designed to flag, assessed by a complementary detection diagnostic, and does their detection ability correspond to their effect on the downstream task?

The remainder of this paper is structured as follows. Section 2 describes the characteristics of the data studied, the selected evaluation metrics, and the evaluated outlier-handling methods (statistical and ML-based). Section 3 presents the experimental methodology and the alignment between the evaluation metrics and the outlier-handling methods described in the previous section. Section 4 reports the downstream-performance and detection results and describes the experimental reproducibility steps. Section 5 reflects on responsible research considerations. Section 6 discusses the findings and their implications, and Section 7 presents conclusions and directions for future work.

2 Background

Using 5G data as a 6G proxy (Tabular and Time-Series): To ground the comparison in realistic 6G-relevant conditions, two real, complementary datasets are used: a tabular dataset of labelled network traffic captured on a functional 5G testbed (5G-NIDD) [20], and a time-series dataset of operational service-level KPIs, CPU and memory utilisation and latency-percentile traces [16]. These were selected because they are publicly available, cover complementary measurement modalities, and, crucially, contain *real* anomalies (labelled attacks and naturally occurring outliers, respectively), so that no synthetic contamination needs to be injected. Although 6G networks are not yet commercially deployed, the use of advanced 5G and Open RAN (O-RAN) measurement data is the established methodological standard for developing 6G machine learning models [14]. The transition from 5G to 6G is fundamentally an architectural evolution reliant on the disaggregated, programmable framework introduced by O-RAN [8]. Because 6G is envisioned as an "AI-native" network, the underlying telemetry structures, tabular traffic and MAC/PHY-layer features, and time-series latency and resource metrics, will remain structurally analogous to current advanced 5G deployments, albeit at higher volumes and lower latencies [17]. Applying denoising and outlier-handling techniques to empirical 5G and edge-computing data therefore provides a representative proxy for 6G network-management challenges. It is acknowledged, however, that results may not transfer directly to true 6G deployments operating at terahertz frequencies and sub-millisecond latency regimes, and this is discussed further in Section 7.

In data-driven network management, measurement data broadly falls into two structural modalities: tabular and time-series. Tabular data consists of independent, multivariate observations where the specific sequence of records is not assumed to carry predictive weight; instead, models rely on the cross-sectional relationships between the features within each individual observation [10]. The 5G-NIDD traffic data is of this form: each record is a flow described by multivariate flow-level features. Conversely, time-series data consists of sequential measurements recorded over continuous intervals, where temporal dynamics, trends, short-term fluctuations, and autocorrelation, are the primary predictive signals [11]; the service-KPI traces are of this form.

2.1 Outlier-Handling Methods

This section describes the cleaning methods evaluated in this study, a no-cleaning baseline (No-Clean) together with three statistical filters and three unsupervised detectors, grouped accordingly. For each method the core algorithm, key hyperparameters, and known limitations relevant to 6G measurement data are discussed. The methods were selected to span a range of interpretability and detection capability: statistical and proximity-based approaches are more interpretable and computationally efficient, whereas the detector family captures multivariate anomaly structure, non-linear in the case of Isolation Forest and LOF, and linear-reconstruction-based in the case of PCA, that per-feature filters cannot capture [5].

Detectors produce an anomaly score for each sample, thresholded at a contamination quantile (tuned per method; Section 3) to flag the most anomalous points. How a flagged point is handled depends on the modality. All **time-series** cleaning is applied *causally*: every statistic, detector, and smoother uses only past and current samples, with parameters estimated on the historical portion of the series, so that no future information leaks into the features used for forecasting. A flagged time step is repaired by *forward-fill*, replaced by the most recent preceding inlier value, a deliberately conservative repair that holds the last good value across the duration of an excursion. In the **tabular** setting a detector flags an entire anomalous *record*. Because over-writing such a record with a single imputed vector while retaining its label would create many identical feature vectors carrying conflicting labels, the repair policy is split by role: in the *training* set, detector-flagged records are **removed** together with their labels, so no contradictory points are introduced into the classifier; in the *test* set, where a prediction is required for every record, they are **imputed** to the inlier column-wise median. The statistical filters do not remove records: Z-score replaces flagged values, IQR clips values to the fence, and Savitzky–Golay smooths every sample; the NoClean baseline leaves the data untouched.

2.1.1 Statistical Baselines

Robust Z-score filtering: Each observation x_i is standardised with a *robust* location and scale estimator,

$$z_i = \frac{|x_i - \tilde{x}|}{1.4826 \cdot \mathrm{MAD}}, \qquad \mathrm{MAD} = \mathrm{median}_j\big(|x_j - \tilde{x}|\big), \tag{1}$$

where $\tilde{x}$ is the sample median and the constant 1.4826 rescales the median absolute deviation to a consistent estimator of the standard deviation under normality. Observations with $z_i > \tau$ are flagged and replaced by the column median (tabular) or, in the causal time-series setting, by the most recent preceding inlier value (forward-fill); the location and scale are estimated on the historical portion of the series, and the threshold τ is the sole hyperparameter. We use the median/MAD form rather than the classical mean/standard-deviation z-score because the latter is sensitive to the very outliers it targets, extreme values inflate μ and σ and mask their own anomaly [2]. The robust estimator removes this masking effect, but the method still assumes an approximately unimodal per-feature distribution and treats each feature independently, ignoring cross-feature and temporal structure.

IQR-based clipping: Outliers are defined as values falling outside the interval

$$\big[Q_1 - k \cdot \mathrm{IQR},\;\; Q_3 + k \cdot \mathrm{IQR}\big], \tag{2}$$

where Q_1 and Q_3 are the first and third quartiles, $\mathrm{IQR} = Q_3 - Q_1$, and k is a scaling multiplier (Tukey's default 1.5). Values outside this range are clipped to the nearest boundary; in the time-series setting the quartiles are estimated on the historical portion so the operation remains causal. As a rank-based method it is more robust to heavy-tailed distributions than Z-score filtering, but it clips *every* value beyond the fence (not only anomalous ones), treats each feature independently, and disregards temporal structure [2].

Savitzky–Golay smoothing (time-series only): For each sample, a polynomial of degree p is fitted by least squares to the *trailing* window of width w ending at that sample, and the sample is replaced by the value of this polynomial at the window's right edge. Because the fit is evaluated at the right edge, the smoother uses only past and current samples and is therefore *causal*, as required so that no future information leaks into the forecasting features. Unlike a simple moving average, the polynomial fit preserves the shape of meaningful signal transients [22]. The two hyperparameters are the window length w (odd, $w > p$) and the polynomial order p, fixed at $p = 3$ following standard practice in signal denoising where cubic fitting balances smoothing against transient-peak preservation.

2.1.2 Unsupervised Detectors

Isolation Forest: Proposed by Liu et al. [15], Isolation Forest exploits the observation that outliers are isolated by random recursive partitioning in fewer splits. An ensemble of T isolation trees is constructed by repeatedly selecting a random feature and a random split value. The anomaly score for a point x is

$$s(x,\, n) = 2^{-\,\mathbb{E}[h(x)]\,/\,c(n)}, \tag{3}$$

where $h(x)$ is the path length of x in a single tree and $c(n)$ is the expected path length of an unsuccessful binary search in a tree of n nodes. Scores close to 1 indicate outliers; scores near 0.5 indicate inliers. Key hyperparameters are the number of trees T and the contamination fraction ρ used to threshold the scores. The method is efficient, makes no distributional assumptions, and scales well to high-dimensional tabular flow data [5].

Local Outlier Factor (LOF): LOF [6] assigns each point a score reflecting how much sparser its local k-nearest-neighbour neighbourhood is compared to those of its neighbours. The local reachability density of a point p is

$$\mathrm{lrd}_k(p) = \left(\frac{1}{k} \sum_{o \,\in\, N_k(p)} \text{reach-dist}_k(p,\, o) \right)^{-1}, \tag{4}$$

where $N_k(p)$ is the set of k nearest neighbours of p and $\text{reach-dist}_k(p, o) = \max\big(d(p, o),\, d_k(o)\big)$, with $d_k(o)$ the k-distance of o. The LOF score is the ratio of the average local reachability density of p's neighbours to that of p itself; values significantly greater than 1 identify outliers. In the causal time-series setting LOF is fitted on the historical data and scores subsequent points out-of-sample.

LOF detects local outliers in regions of varying density, but its $\mathcal{O}(n^2)$ complexity limits scalability and performance is sensitive to the choice of k [5].

PCA reconstruction: Principal Component Analysis provides a *linear* reconstruction detector, i.e. a linear autoencoder. The (standardised) data are projected onto the k leading principal directions V_k and reconstructed; the residual to this low-dimensional subspace serves as the anomaly score,

$$\hat{\mathbf{x}}_i = V_k V_k^\top \tilde{\mathbf{x}}_i, \qquad s(\mathbf{x}_i) = \left\|\tilde{\mathbf{x}}_i - \hat{\mathbf{x}}_i\right\|_2, \tag{5}$$

where $\tilde{\mathbf{x}}_i$ is the standardised input and V_k holds the top-k loadings. Points lying far from the principal subspace reconstruct poorly and receive high scores [12]. The single structural hyperparameter is the number of components k, with scores thresholded at a contamination quantile. PCA is deterministic, has negligible runtime, and is well matched to the moderate dimensionality of the data here (the tabular flow features and the multivariate KPI vector of the series), where a deep neural autoencoder would be over-parameterised. The cost of this choice is expressive: being linear, PCA captures only linear correlation structure and cannot model the non-linear manifolds a deep autoencoder could; we therefore treat it as a lightweight reconstruction baseline and regard deep reconstruction variants as out of scope. For tabular data each flow record is the input $\mathbf{x}_i$; for time-series data each input is the multivariate KPI vector at a time step, so the same detector applies to both modalities. In the causal time-series setting the projection V_k is fitted on the historical data and applied to subsequent points.

2.2 Evaluation Metrics

Because neither dataset provides an outlier-free clean reference, the 5G-NIDD attacks are real events and the KPI outliers are genuine measurements, not injected corruptions, fidelity to a known clean signal cannot be measured. The evaluation is therefore *downstream-centric*: every method is judged by its effect on the downstream task, complemented by a detection diagnostic and a runtime measurement.

2.2.1 Downstream Performance Metrics

Macro-F1 and ROC-AUC (classification): For the tabular task, distinguishing attack from benign traffic, downstream performance is measured by the macro-averaged F1 score and the area under the ROC curve (ROC-AUC). Macro-F1 computes the F1 score independently for each class and averages them with equal weight, so performance on both classes counts equally regardless of their relative frequency; this matters in a security setting, where both missed attacks and false alarms carry operational cost, and where accuracy alone can be misleading on uneven class distributions [24]. ROC-AUC complements it as a threshold-independent measure of how well a classifier ranks attacks above benign traffic, decoupling the comparison from any particular decision threshold.

RMSE and MAE (regression): For the time-series task, one-step-ahead forecasting of the 99th-percentile latency (`lat99`) from the current multivariate KPI vector, performance is measured by both root mean square error (RMSE) and mean absolute error (MAE),

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \qquad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{6}$$

where $\hat{y}_i$ and y_i are the predicted and observed latency values on the held-out test segment of each cross-validation fold. RMSE is the primary metric because its quadratic penalty assigns disproportionately higher cost to large latency errors, consistent with the operational reality of 6G network management, where tail-latency spikes are more disruptive than uniformly distributed small errors. MAE is reported alongside RMSE as a *complementary diagnostic* rather than as independent corroboration, since the two are strongly correlated. As Willmott and Matsuura establish, RMSE is a function of both the average error magnitude and the variability of the error distribution, making it ambiguous in isolation [25]. A substantial RMSE–MAE divergence across methods indicates a method that handles large latency spikes differently from the smoothly varying background; convergence indicates that the residual errors are uniformly distributed.

2.2.2 Detection Diagnostic

Because the tabular dataset carries ground-truth anomaly labels, each unsupervised detector is additionally assessed *directly*, independently of any downstream cleaning, by treating its anomaly score as a soft classifier of the attack label and reporting ROC-AUC and the area under the precision–recall curve (PR-AUC); a ROC-AUC of 0.5 corresponds to chance. This diagnostic answers a question distinct from downstream utility: whether the statistical-outlier notion these detectors encode actually aligns with the real anomalies present in the data.

2.2.3 Runtime

The wall-clock execution time of each method's cleaning step is recorded as the mean over all random seeds (and, for the time-series modality, the blocked cross-validation folds), reported in milliseconds. This is motivated by the deployment context: 6G network-management systems operate under strict latency budgets that vary by control tier. At the tightest extreme, 6G URLLC targets end-to-end delays below 0.1 ms [9], while the O-RAN near-real-time RIC, the architectural layer most directly relevant to KPI-driven cleaning pipelines, operates on a control-loop budget of 10 ms to 1 s [1]. A data-cleaning step must therefore complete well within the lower bound of this window to leave headroom for the inference, actuation, and communication steps that follow it. A method that achieves marginally superior downstream performance at substantially greater computational cost may be operationally unsuitable regardless of its statistical merits; results in Section 4 are interpreted against the 10 ms near-RT RIC threshold as a concrete operational ceiling. These runtimes are wall-clock measurements taken on a single machine and are interpreted as orders of magnitude rather than precise benchmarks.

3 Downstream-Centric Evaluation Design

This section specifies the experimental design used to compare the cleaning methods of Section 2. Four choices define it: every method is run against an explicit NoClean baseline; every method is evaluated through two downstream models of deliberately contrasting noise-sensitivity; every hyperparameter is selected by non-oracle tuning that never sees the test labels; and every comparison against NoClean is a paired significance test with false-discovery-rate (FDR) correction. Because neither dataset offers an outlier-free reference (Section 2.2), no method is judged by fidelity to a clean signal: the verdict is always the downstream task, complemented on the labelled tabular data by the detection diagnostic of Section 2.2. The same design is instantiated on each modality with a task-appropriate protocol.

Reporting two probes, rather than a single reference model, is the central design choice. The Random Forest is a non-parametric, scale-invariant ensemble whose bagging over bootstrapped trees makes it *robust* to a small fraction of corrupted points and thus able to absorb noisy inputs, whereas the distance-weighted k-NN ($k = 15$, standardised features) depends directly on feature-space distances and is correspondingly *sensitive* to them. Contrasting the two tests whether any benefit of cleaning is conditional on the downstream model's noise-sensitivity rather than assuming it, and using the same pair across both modalities, as classifiers for the tabular task and regressors for the time-series task, keeps performance differences attributable to the cleaning methods rather than to an arbitrary model choice.

On the tabular 5G-NIDD data the task is to separate attack from benign flows using their numeric flow-level features. For each seed we draw a class-balanced subsample of 10,000 flows and split it 80/20 in a stratified manner; the training and test features are cleaned under the role-split repair policy of Section 2, and a Random Forest and a k-NN classifier are trained on the cleaned training set and scored by macro-F1 and ROC-AUC against the true labels. Because this dataset carries ground-truth labels, each detector is additionally assessed once per seed by the detection diagnostic of Section 2.2, independently of any downstream cleaning, which isolates whether the outlier notion the detectors encode corresponds to the real attacks at all, a question downstream accuracy cannot answer on its own.

On the time-series data the task is one-step-ahead forecasting of the 99th-percentile latency (`lat99`) from the current 15-channel KPI vector. All cleaning here is causal and flagged steps are repaired by forward-fill (Section 2), so that no future information leaks into the forecasting features; a Random Forest or k-NN regressor is then trained on the cleaned history and scored on the held-out tail by RMSE (primary) and MAE. In both modalities NoClean operates on the raw inputs.

To keep the comparison honest, each method's hyperparameters are selected per run by Optuna (TPE, seeded) under an objective that uses no clean ground truth and mirrors the deployment repair policy: on the tabular task it drops flagged records in the training fold and imputes the validation fold, scoring macro-F1, while on the time-series task it cleans an inner training segment causally and scores RMSE on its held-out tail. No method is therefore tuned against information it would not possess at deployment, and the k-NN neighbour count is held fixed so the probe remains a constant yardstick across methods. The unit of replication is chosen to vary the underlying data rather than only model randomness. On the tabular task each of five seeds independently re-subsamples and re-splits the dataset, so the seed is itself the replicate ($n = 5$). On the time-series task a single fixed split would tie the result to one window of the series, so `lat99`, which is approximately stationary, is evaluated with *blocked cross-validation*: the series is partitioned into five disjoint contiguous segments, and within each an independent causal forecast (an internal 80/20 split, cleaning fitted only on that segment's history, and prediction of the held-out tail) is run, making the replicate the (seed, fold) pair and giving $5 \times 5 = 25$ replicates whose paired comparisons span different temporal regimes. Across these replicates every cleaner is compared against NoClean on identical data with a paired t-test, controlling the false-discovery rate over each metric's comparison grid with the Benjamini–Hochberg procedure (Section 4).

4 Experimental Setup and Results

We evaluate the cleaning methods of Section 2 (a NoClean baseline, three statistical filters, and three unsupervised detectors) under the downstream-centric, dual-probe (Random Forest / k-NN) protocol of Section 3, on two real datasets with no synthetic injection. This section gives the datasets and environment, the statistical procedure, and the downstream, detection, and runtime findings.

The tabular dataset is 5G-NIDD [20], a fully labelled network-intrusion dataset captured on a functional 5G testbed (University of Oulu), in which attacker nodes launch DoS floods and port scans against a server amid benign 5G user traffic. Each record is a bidirectional flow described by Argus flow-level features; we use the numeric features (39 after dropping identifiers and the attack-type/tool columns that would leak the label), with the binary attack/benign label. The full release is ≈61% attack; per seed we draw a class-balanced subsample of 10,000 flows and split it 80/20 (stratified).

The time-series dataset is a regularised trace of operational service KPIs [16] (CPU/memory utilisation and latency percentiles), from which we forecast the 99th-percentile latency (`lat99`) one step ahead from the current multivariate KPI vector. The series is approximately stationary, so blocked cross-validation is applied directly; the naturally occurring heavy-tailed per-column outlier from the analysis performed in preprocessing (latency percentiles with ∼1.4% of points beyond 3σ) provide the anomalies to be handled. Experiments ran on a single machine with an Intel Core i5-11300H CPU under Ubuntu 24.04 on WSL2; all computation is CPU-only (scikit-learn, NumPy/SciPy). Code is Python 3.12.3 with NumPy 2.4.4, pandas 3.0.2, SciPy 1.17.1, scikit-learn 1.8.0, Matplotlib 3.10.9, and Optuna 4.9.0. Random Forests are fit with `n_jobs`$= -1$; per-method cleaning runtime is measured with `time.perf_counter()` (code open-sourced at: GitHub).

Cleaning hyperparameters are selected per run by non-oracle Optuna (TPE, seeded; Section 3). The replicate is the seed for the tabular task (five seeds, each re-subsampling and re-splitting the data; $n = 5$) and the (seed, fold) pair for the time-series task (five seeds $\times$ five blocked folds; $n = 25$). Because every method sees identical data at a given replicate, each cleaner is compared to NoClean with a per-replicate paired t-test; the false-discovery rate is controlled across each metric's comparison grid with the Benjamini–Hochberg (BH) procedure. A result is *significant* only if it improves on NoClean with $q < 0.05$ (*); † marks effects significant before correction only. At the tabular $n = 5$ the Wilcoxon signed-rank test cannot reach $\alpha = 0.05$ and is omitted.

Downstream: tabular attack classification Table 1 and Figure 1 report macro-F1 and ROC-AUC (the full per-method significance is given in Appendix A, Table 4). **No cleaning method improves on NoClean for either probe** (0/10 comparisons improve; none reach $q < 0.05$ as an improvement). NoClean and the robustly-tuned Z-score filter are statistically indistinguishable and jointly best (macro-F1 0.738–0.739, ROC-AUC 0.853–0.857); the multivariate detectors (IF, LOF, PCA) sit marginally below NoClean; and **IQR significantly *degrades* both probes** (macro-F1 0.710–0.711, ROC-AUC 0.81–0.82; $q = 0.009$), as global winsorising clips legitimate flow features. The result is a clean null: on real 5G traffic, outlier handling yields no downstream classification gain, and aggressive clipping is actively harmful.

Table 1: Tabular attack classification on 5G-NIDD (mean ± std over 5 seeds; higher is better). Best per probe in **bold**. No cleaner improves on NoClean (0/10, BH-FDR); IQR is significantly *below* NoClean for both probes ($q = 0.009$).

Probe	Method	Macro-F1	ROC-AUC	Runtime (ms)
RF	NoClean	**0.738 ± 0.010**	**0.857 ± 0.009**	0.3
	Z-score	0.738 ± 0.010	0.857 ± 0.009	10.1
	IQR	0.710 ± 0.010	0.814 ± 0.011	9.3
	Isolation Forest	0.735 ± 0.011	0.854 ± 0.009	278.4
	LOF	0.735 ± 0.009	0.854 ± 0.009	105.9
	PCA	0.734 ± 0.009	0.853 ± 0.008	13.6
kNN	NoClean	**0.739 ± 0.010**	**0.853 ± 0.012**	0.3
	Z-score	0.738 ± 0.010	0.853 ± 0.012	10.3
	IQR	0.711 ± 0.009	0.821 ± 0.011	8.3
	Isolation Forest	0.737 ± 0.008	0.852 ± 0.010	267.8
	LOF	0.736 ± 0.010	0.850 ± 0.013	106.0
	PCA	0.737 ± 0.010	0.853 ± 0.012	46.6

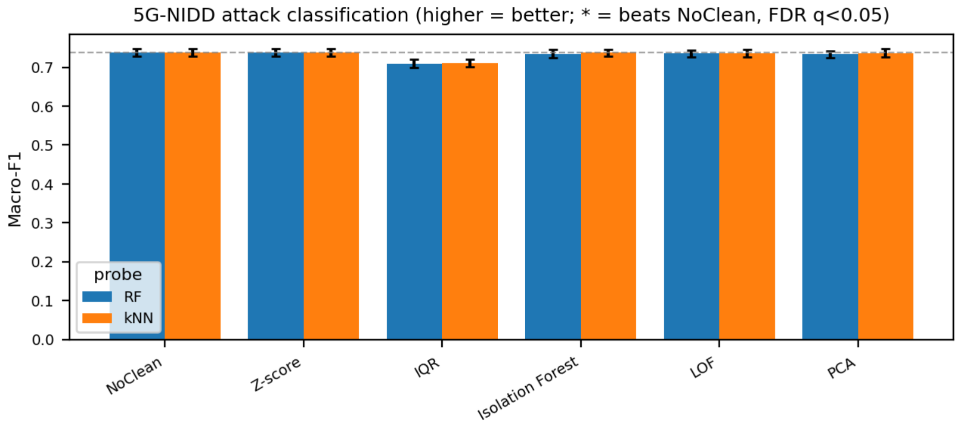

Figure 1: Tabular attack-classification macro-F1 by method and probe (bars: mean; whiskers: std over 5 seeds; dashed line: NoClean). No cleaner significantly exceeds NoClean for either probe.

Detection diagnostic: Scored directly against the attack labels, the unsupervised detectors rank attacks only marginally above chance (Table 2; ROC-AUC 0.535–0.598), despite a supervised Random Forest separating the same classes at ROC-AUC ≈ 0.86 (Table 1). The signal is therefore present but *not* of the form statistical outlier detectors assume: the attacks (a $\sim$61% majority dominated by dense, repetitive DoS floods) are not rare, isolated points. This is the central reason cleaning does not help the tabular task, statistical outlier detection is a poor proxy for intrusion detection here.

Table 2: Detection diagnostic: unsupervised detector score vs. the attack label (mean $\pm$ std over 5 seeds). ROC-AUC 0.5 is chance.

Detector	ROC-AUC	PR-AUC
Isolation Forest	0.598 $\pm$ 0.002	0.579 $\pm$ 0.003
LOF	0.541 $\pm$ 0.005	0.523 $\pm$ 0.004
PCA reconstruction	0.535 $\pm$ 0.018	0.536 $\pm$ 0.010

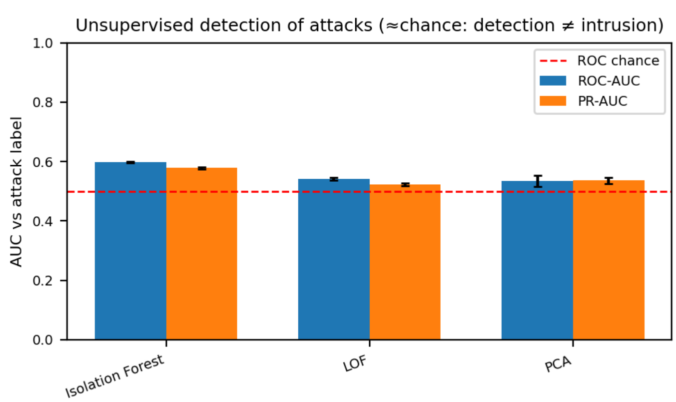

Figure 2: Unsupervised detector ROC-AUC and PR-AUC against the real attack label; the dashed line marks chance (ROC-AUC = 0.5). All three detectors are close to chance.

Downstream: time-series forecasting Table 3 reports forecasting error (full RMSE and MAE significance in Appendix A, Tables 5–6). **After FDR correction, no method significantly improves on NoClean** (0/12 RMSE and 0/12 MAE comparisons reach $q < 0.05$ as an improvement). **Savitzky–Golay smoothing is the only *suggestive* gain**: under the RF probe it lowers RMSE by $\approx 17\%$ ($6878 \to 5678$) and wins on 21/25 replicates ($p = 0.006$), and lowers MAE on 23/25 ($p = 0.015$), but neither survives correction ($q = 0.067$, 0.057); PCA under RF is similarly suggestive ($q = 0.078$). The density/partition detectors are neutral-to-harmful (Isolation Forest and LOF *raise* RMSE under both probes), and the point-removal filters give no reliable benefit. Crucially, the absolute RMSE carries very large variance (std comparable to the mean) because the blocked folds span $\sim$two orders of magnitude in difficulty; the paired improvement-over-NoClean view (Fig. 3), which cancels this between-fold variance, shows the same ordering, Savitzky–Golay slightly positive, detectors negative, and is the basis for the significance tests above.

Table 3: Time-series `lat99` forecasting error (mean $\pm$ std over $5 \times 5 = 25$ replicates; lower is better). Best per probe in **bold**. After BH-FDR, no method significantly improves on NoClean; † marks effects significant before correction only (Savitzky–Golay / RF).

Probe	Method	RMSE	MAE
RF	NoClean	6878 ± 5205	4414 ± 3915
	Z-score	6928 ± 5308	4458 ± 3961
	IQR	6857 ± 5234	4414 ± 3922
	Savitzky-Golay	**5678 ± 3519**†	**3655 ± 2753**†
	Isolation Forest	7134 ± 5397	4705 ± 4365
	LOF	7651 ± 4829	5209 ± 3634
	PCA	6307 ± 4262	4159 ± 3230
kNN	NoClean	5398 ± 2481	**2893 ± 1421**
	Z-score	5436 ± 2544	2927 ± 1503
	IQR	5431 ± 2549	2918 ± 1475
	Savitzky-Golay	**5392 ± 2503**	3008 ± 1555
	Isolation Forest	6784 ± 5212	3922 ± 3488
	LOF	5914 ± 2572	3355 ± 1793
	PCA	5548 ± 2433	3045 ± 1419

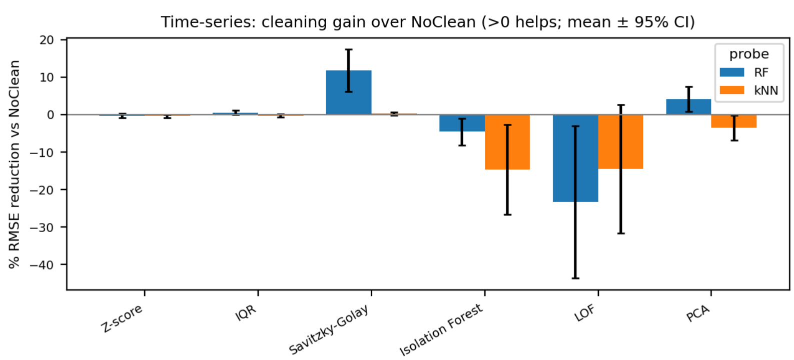

Figure 3: Time-series forecasting: paired improvement over NoClean (% RMSE reduction per (seed, fold) replicate; mean $\pm$ 95% CI). Values > 0 indicate the cleaner beats NoClean. This paired view cancels the large between-fold variance of the absolute RMSE.

Runtime: The cleaning step's wall-clock cost (rightmost column of Table 1; Fig. 4) separates the methods against the ∼10 ms near-RT RIC budget. NoClean, the statistical filters, Savitzky–Golay, and PCA are within budget (NoClean < 1 ms; Z-score/IQR/Savitzky–Golay ≈ 1–10 ms; PCA ≈ 3–47 ms), whereas the density/partition detectors are one-to-two orders of magnitude slower (LOF ≈ 40–106 ms, Isolation Forest ≈ 270–280 ms) and are not deployable in the near-RT loop. Notably, the *only* method showing any (suggestive) downstream benefit, Savitzky–Golay, is also among the cheapest (≈ 1 ms), whereas the costliest methods (IF, LOF) are also the most harmful. (Statistical significance of these results was not tested as the runtime is highly dependable on specific hardware and the focus is not on exact values, but rather the orders of magnitude.)

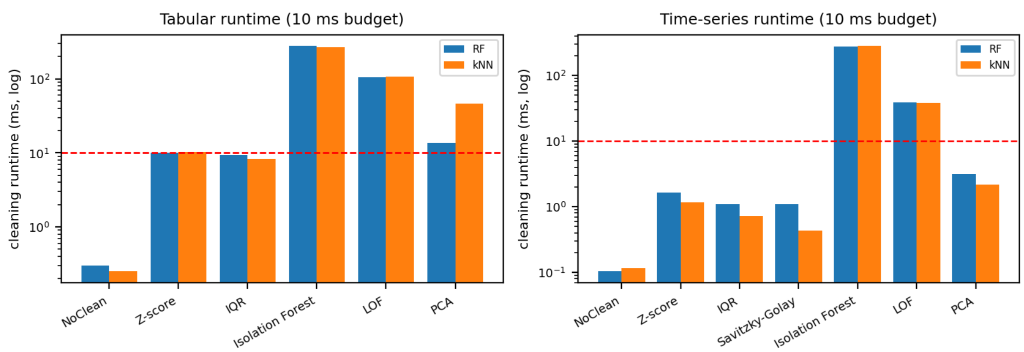

Figure 4: Per-method cleaning runtime (log scale, ms) for the tabular and time-series modalities; the dashed line marks the 10 ms near-RT RIC budget.

5 Responsible Research

This study was conducted in accordance with the Netherlands Code of Conduct for Research Integrity [13], and its design reflects that code's principles of honesty, transparency, and reproducibility more than any single declaration could. Because the work is methodological and draws only on telemetry from a controlled testbed and an instrumented service , network flows and system KPIs, not data about people , it raises no human-subject or personal-data concerns. No personally identifiable or sensitive information is processed, and no consent or anonymisation is therefore required; indeed, on the tabular side we retain only numeric flow-level features and discard the identifier columns outright, both to prevent label leakage and because such identifiers carry no analytic value. The substantive responsibilities instead lie in data provenance, reproducibility, honest reporting, and the careful scoping of claims.

On provenance, both datasets are publicly released and are reused strictly within their intended scope, with the original sources cited. The tabular data is the 5G-NIDD network-intrusion dataset [20], captured on a functional 5G testbed and comprising labelled attack and benign flows; the time-series data consists of operational service KPIs , CPU and memory utilisation and latency percentiles , derived from the web-server performance traces of Mekki et al. [16], which we use after a documented preprocessing step that regularises them to a 6 s grid and robust-scales them. Crucially, we introduce no synthetic contamination: every anomaly analyzed is one already present in the released data , labeled attacks in the tabular case and naturally occurring heavy-tailed outliers in the time-series case , so the evaluation reflects real, rather than fabricated, corruption.

Transparency and honesty were built into the evaluation itself rather than asserted after the fact. The pipeline is reproducible end-to-end: tabular results are averaged over five fixed random seeds and time-series results over five seeds across five blocked cross-validation folds; the sub-sampling and splitting protocol, the non-oracle hyperparameter-selection procedure, the blocked cross-validation scheme, and the exact software environment are all specified; the computation runs entirely on commodity CPU hardware; and the implementation, together with the generated tables and figures, is released for independent regeneration. The same discipline guards against overstated findings. Hyperparameters are selected by an objective that never sees the test labels or any clean ground truth, so no method , least of all the detectors , is granted an unrealistic "oracle" advantage; all time-series cleaning is causal, so no future information leaks into the forecasting features; every method is measured against an explicit no-cleaning baseline; and each claimed improvement is subjected to a paired significance test with false-discovery-rate correction. Where this produced a negative result, that no method significantly improves the downstream task on *either* modality after correction, and that the unsupervised detectors identify the real attacks only marginally above chance, we report it plainly rather than searching for a favorable comparison, and we label the single borderline effect (Savitzky–Golay smoothing on the forecasting task) as suggestive rather than significant.

Finally, we are deliberate about the scope of the resulting claims and their likely use. The intended benefit of this work is practical: to help practitioners direct limited data-preparation and compute effort to where it measurably improves automated network management, thereby avoiding unnecessary processing and supporting more dependable downstream decisions such as fault detection and resource allocation. Because those decisions can affect service quality, we are careful not to over-generalise. The conclusions are bounded by the use of one dataset per modality (and, for the time-series, a single series), a class-balanced tabular subsample, a one-step forecasting horizon, and a 5G/edge proxy for not-yet-deployed 6G systems , limitations we set out explicitly in Section 6. In particular, the recommendation to omit a cleaning stage for robust models is a finding about the conditions studied, not a universal rule: it should be validated on the target deployment before being acted upon, since a choice that is harmless here could matter under a different noise regime or in a safety-relevant task. The methods and datasets carry no notable dual-use risk, and the open release of the code (open-sourced at: GitHub) and analysis is intended to support transparent, community-verifiable research rather than any harmful application.

6 Discussion

The central finding is a *double null*: under a leakage-free, non-oracle evaluation on real 5G data with no synthetic contamination, no cleaning method reliably improves the downstream task on either modality. On the tabular attack-classification task, no method beats NoClean for either probe (0/10 comparisons), and the IQR filter is significantly *worse* ($q = 0.009$; Section 4). On the time-series forecasting task, after Benjamini–Hochberg correction no method is significant either (0/12 RMSE and 0/12 MAE comparisons); the only suggestive benefit is Savitzky–Golay smoothing, which lowers RF RMSE by $\approx 17\%$ and wins on 21/25 replicates but does not survive correction ($q = 0.067$; Section 4). The result holds across both the robust Random Forest and the sensitive k-NN probe, so it is not an artefact of the downstream model. Rather than a contest between cleaning algorithms, the experiments suggest a more basic point: outlier handling helps only when the corruption is genuinely separable noise, and on these datasets the anomalies are largely real, structured signal that generic cleaning discards rather than recovers.

This is seen most clearly in the tabular task through the detection diagnostic (Section 4). The unsupervised detectors rank the real attacks barely above chance (ROC-AUC 0.535–0.598), even though a supervised Random Forest separates the same classes at ROC-AUC ≈ 0.86. The discriminative signal is therefore present, but it is not of the form statistical outlier detection assumes: the 5G-NIDD attacks are a dense majority dominated by repetitive DoS floods, not rare isolated points. Statistical outlier detection is thus a poor proxy for intrusion detection here, and "cleaning" by removing or imputing flagged records simply deletes informative samples, which is why the detectors sit marginally below NoClean and IQR, by winsorising legitimate heavy-tailed flow features, degrades performance outright. The robustness of Random Forest's bagging and of distance-weighted k-NN to a small fraction of corrupted points leaves cleaning little to add, so NoClean is competitive or best throughout.

The time-series task is where cleaning comes closest to helping, and the mechanism is instructive. `lat99` is a heavy-tailed signal whose extremes are genuine, informative events. Methods that delete or clip values (Z-score, IQR) or that flag and repair points (Isolation Forest, LOF) tend to discard or flatten real signal and are neutral-to-harmful, Isolation Forest and LOF *raise* RMSE under both probes. Savitzky–Golay smoothing behaves differently: it attenuates short-term noise while preserving the location of true transients, which is why a simple, interpretable, causal filter is the only method with a (suggestive) forecasting gain, and why low-rank PCA reconstruction is the next closest. That even smoothing fails FDR correction reflects both a modest effect and the large between-fold variance of this single series: the blocked folds span roughly two orders of magnitude in forecasting difficulty, so absolute error is dominated by which temporal regime a fold lands in rather than by the cleaning method. The paired improvement-over-NoClean view (Fig. 3) cancels this variance and makes the ordering, smoothing slightly positive, detectors negative, visible, but the magnitude is not large enough to clear correction at $n = 25$.

A recurring theme across both modalities is that a method's notion of "outlierness" does not align with what actually helps the task. The detectors assign anomaly scores that neither match the labelled attacks (tabular) nor improve the forecast (time-series); where they do act, removal and repair cost more in lost information than they recover. This argues for treating downstream task performance as the primary verdict and the detection diagnostic as exactly that, a diagnostic, rather than selecting a cleaning method by its outlier scores or by distributional similarity to an assumed clean reference, which in this real-data setting does not exist.

Combining accuracy with runtime sharpens the practical reading (Section 4). The statistical filters, Savitzky–Golay, and PCA all complete within the $\sim$10 ms near-RT RIC control budget ($\lesssim$10 ms), whereas LOF and Isolation Forest are one-to-two orders of magnitude slower ($\approx$ 40–280 ms) and fall outside that loop. The two facts compound unfavourably for the detectors: the methods that are most expensive (Isolation Forest, LOF) are also the most harmful, while the only method with any suggestive benefit (Savitzky–Golay) is among the cheapest ($\approx$ 1 ms). For these real 5G workloads the practical recommendation is therefore conservative: a lightweight causal smoother is worth trying on noisy time-series KPI streams, but a generic outlier-handling stage, and the unsupervised detectors in particular, is hard to justify, consuming compute (and sometimes degrading accuracy) without measurable benefit.

Several factors bound the scope of these conclusions. First, the evaluation uses *one dataset per modality* and a 5G/edge proxy for 6G, so quantitative results need not transfer to terahertz, sub-millisecond 6G telemetry; and for the time-series, the blocked folds vary windows of a *single* series, so the significance generalises across temporal regimes of that series rather than across independent series. Second, the tabular subsample is class-balanced rather than at the dataset's natural $\sim$61% attack prevalence, so the reported macro-F1 reflects a balanced operating point; and the attacks studied are DoS- and scan-dominated, so the detector–attack mismatch may be less severe for stealthier, rarer intrusions that better fit the outlier assumption. Third, with five seeds the tabular paired tests have limited power, and the 25 time-series replicates, while more numerous, come from one series and overlap in difficulty, so the absence of significance should be read as "no reliable effect detected" rather than proof of no effect. Fourth, the forecasting task is one-step-ahead from the current multivariate KPI vector; longer horizons or richer lag structure could change the value of cleaning. Finally, the repair policies are deliberately simple, detector-flagged records are dropped at training and median-imputed at test, time-series points are forward-filled, and PCA reconstruction is linear, so the detector and reconstruction results should be read as bounds for these specific, lightweight choices rather than as verdicts on all detection or reconstruction approaches. Whether a stronger repair (e.g. model-based imputation) or non-linear reconstruction would change the tabular null is left to future work (Section 7).

7 Conclusions and Future Work

This work asked how denoising and outlier-handling methods compare in preserving the usefulness of noisy 6G network measurements for downstream machine-learning tasks across tabular and time-series modalities, and which strategy best suits the dominant anomalies of each modality. To answer this without resorting to synthetic corruption, we evaluated on two *real, labelled* datasets attack traffic from a functional 5G testbed (5G-NIDD) and a heavy-tailed operational latency series used for short-term forecasting and compared six cleaning methods, from simple interpretable filters (Z-score replacement, IQR clipping, Savitzky–Golay smoothing) to unsupervised detectors (Isolation Forest, Local Outlier Factor, and PCA reconstruction), against a no-cleaning baseline. Hyperparameters were chosen by non-oracle tuning that never sees the test labels, each method was judged under two contrasting downstream probes (a robust Random Forest and a sensitive k-NN), and every claimed improvement was subjected to a paired significance test with multiple-comparison correction. We organised the investigation around three subquestions, which the experiments answer as follows.

The first two concern the downstream comparison. On *whether* cleaning helps and *which* method is best (SQ1), the answer is largely negative: after false-discovery-rate correction, no method significantly improved on doing nothing on either modality (0/10 tabular and 0/12 time-series comparisons). On the tabular attack-classification task the no-cleaning baseline tied with a robustly-tuned Z-score filter was the best choice, and aggressive clipping (IQR) significantly *degraded* performance; on the time-series forecasting task the only method to help at all was lightweight Savitzky–Golay smoothing ($\approx 17\%$ lower Random-Forest error, winning on 21/25 replicates), but its gain did not survive correction. On *whether this benefit is conditional* (SQ2), the effective strategy differs by modality no cleaning for the tabular classifier, at most a causal smoother for the heavy-tailed series but it does *not* depend on the downstream model: the same ordering and the same absence of a reliable gain hold under both the robust Random Forest and the noise-sensitive k-NN. The deliberately sensitive probe, included precisely to expose any latent advantage of cleaning, revealed none, indicating that the null is a property of the data and tasks rather than of an arbitrary model choice.

The third subquestion (SQ3) asks whether the unsupervised detectors actually identify the real, labelled anomalies they are designed to flag. They do not: scored directly against the ground-truth labels, the detectors ranked attacks only marginally above chance (ROC-AUC 0.54–0.60), even though a supervised Random Forest separated the same classes at ≈ 0.86. Detection ability also failed to translate into downstream usefulness where the detectors flagged and repaired points, removal cost more in lost information than it recovered. Because the 5G-NIDD attacks are a dense, DoS-dominated majority rather than rare isolated points, statistical outlier detection is a poor proxy for intrusion detection here, and a method's notion of "outlierness" simply did not align with what helped the task.

Taken together, these answers give a clear if negative message and a conservative recommendation: *omit a generic outlier-handling stage by default*, judging any candidate by downstream performance rather than by anomaly scores or distributional similarity, with the single exception of a lightweight causal smoother on heavy-tailed sequential signals. Runtime reinforces this: the cheap methods (the statistical filters, Savitzky–Golay, and PCA on the series) fall within the $\sim$10 ms near-real-time control budget, whereas Local Outlier Factor and Isolation Forest are one-to-two orders of magnitude slower and, here, also the most harmful so the costliest methods are precisely the ones a practitioner can most safely drop. Beyond this headline, the study's contribution is methodological: an evaluation framing real anomalies rather than injected ones, non-oracle tuning, an explicit no-cleaning baseline, replication across seeds and blocked temporal folds, and multiple-comparison correction that separates reproducible gains from noise and lets a negative result be stated with confidence.

Several open issues bound these conclusions and motivate further work. First, external validity: the study uses one dataset per modality and, for the time-series, a single series partitioned into temporal folds, so the absence of a reliable cleaning benefit should be confirmed across more datasets and independent series before it is generalised. Second, the repair policies were deliberately simple detector-flagged records were dropped at training and median-imputed at test, time-series points were forward-filled, and PCA reconstruction is linear so a natural question is whether a stronger, non-destructive repair (model-based or k-NN imputation) or a sequence-aware neural reconstruction (recurrent or convolutional autoencoders), able to exploit temporal structure that linear PCA and smoothing cannot, would overturn the tabular null or surpass simple smoothing on heavy-tailed signals. Third, the attacks studied are dense and DoS-dominated; testing stealthier, rarer intrusions that better fit the outlier assumption would establish whether unsupervised detection ever becomes the right tool. Finally, because cleaning competes for the same millisecond-scale budget as the model it feeds, jointly optimising it against end-to-end 6G control-loop latency rather than treating accuracy and runtime separately is a promising direction. These results do not claim that data cleaning is never worthwhile; they show that, for two realistic 6G-relevant tasks, its benefit must be demonstrated rather than assumed.

References

[1] Oscar Adamuz-Hinojosa, Lanfranco Zanzi, Vincenzo Sciancalepore, Andres Garcia-Saavedra, and Xavier Costa-Pérez. ORANUS: Latency-tailored orchestration via stochastic network calculus in 6G O-RAN. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*. IEEE, 2024. arXiv:2401.03812.

[2] Herman Aguinis, Ryan K. Gottfredson, and Harry Joo. Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods*, 16(2):270–301, 2013.

[3] Babak Azkaei, Kishor Chandra Joshi, and George Exarchakos. Machine learning-driven anomaly detection for 5G O-RAN performance metrics. In *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2025. arXiv:2509.03290.

[4] Mohammed Banafaa et al. 6G mobile communication technology: Requirements, targets, applications, challenges, advantages, and opportunities. *Alexandria Engineering Journal*, 64:245–274, 2023.

[5] Azzedine Boukerche, Lining Zheng, and Omar Alfandi. Outlier detection: Methods, models, and classification. *ACM Computing Surveys*, 53(3):1–37, 2020.

[6] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In *ACM SIGMOD Record*, volume 29, pages 93–104, 2000.

[7] E. Edozie, A. N. Shuaibu, B. O. Sadiq, and U. K. John. Artificial intelligence advances in anomaly detection for telecom networks. *Artificial Intelligence Review*, 58(4), January 2025.

[8] Sneihil Gopal, Aziz Kord, and Richard A. Rouil. O-ran and 6g: The future of wireless innovation?, 2024.

[9] Abdikarim Mohamed Ibrahim, Rosdiadee Nordin, Yahya S.M. Khamayseh, Angela Amphawan, and Muhammed Basheer Jasser. URLLC for 6G enabled industry 5.0: A taxonomy of architectures, cross layer techniques, and time critical applications. *arXiv preprint arXiv:2510.08080*, 2025.

[10] Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6):6488–6508, 2026.

[11] Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey, 2025.

[12] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, apr 2016.

[13] KNAW, NFU, NWO, TO2, VH, and VSNU. Netherlands code of conduct for research integrity. Technical report, Association of Universities in the Netherlands, 2018.

[14] Z. Li, Q. Wang, Y. Wang, and T. Chen. The architecture of AI and communication integration towards 6G: An O-RAN evolution. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 2329–2334, December 2024.

[15] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 413–422, 2008.

[16] Mohamed Mekki, Nassima Toumi, and Adlen Ksentini. Benchmarking on microservices configurations and the impact on the performance in cloud native environments, 2022. Dataset.

[17] M. Owais and T. Shongwe. Machine learning-enabled 5G and 6G networks: Methods, challenges, and opportunities. *Applied Sciences*, 16(4):2071, February 2026.

[18] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, 34(3):134–142, 2019.

[19] Mamoon M. Saeed, Rashid A. Saeed, Maha Abdelhaq, Raed Alsaqour, Mazin K. Hasan, and Rania A. Mokhtar. Anomaly detection in 6G networks using machine learning methods. *Electronics*, 12(15):3300, 2023.

[20] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, and Mika Ylianttila. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network, 2022.

[21] S. Sankar Ganesh and S. Janakiraman. VARNet-6G with FIERO model for anomaly detection and enhancing network stability in future-ready communication systems. *Scientific Reports*, 15(1):34390, 2025.

[22] Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964.

[23] Anshuman Shrivastava et al. A survey on anomaly detection in IoT: Techniques, challenges, and opportunities with the integration of 6G. *Computer Networks*, 2025.

[24] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, July 2009.

[25] Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005.

[26] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

A Full paired-significance tables

For completeness, Tables 4–6 give the full per-method significance underlying Section 4. Each cleaner is compared to NoClean on the identical data at every replicate — five seeds for the tabular task ($n = 5$) and the $5 \times 5 = 25$ (seed, fold) pairs for the time-series task ($n = 25$) — with a paired t-test and Benjamini–Hochberg (BH) control of the false-discovery rate over each metric's comparison grid. Δ is the mean improvement over NoClean (positive means the cleaner is better: higher macro-F1, or lower error); "wins" counts replicates in which the cleaner beats NoClean; p is the uncorrected and q the BH-corrected significance. **No cleaner achieves a significant improvement ($q < 0.05$, $\Delta > 0$) on any metric.** † marks an improvement significant before correction only ($p < 0.05$, $q \geq 0.05$); $^{\downarrow}$ marks a significant *degradation* ($q < 0.05$, $\Delta < 0$). Time-series Δ values are in the latency units of Table 3.

Table 4: Tabular attack-classification macro-F1: each cleaner vs. NoClean ($n = 5$ seeds).

Probe	Method	ΔF1	Wins	p	q
RF	Z-score	−0.0002	0/5	0.178	0.222
	IQR	−0.0281	0/5	0.002	$0.009^{\downarrow}$
	Isolation Forest	−0.0031	0/5	0.017	0.058
	LOF	−0.0024	0/5	0.039	0.098
	PCA	−0.0040	0/5	0.049	0.098
kNN	Z-score	−0.0005	1/5	0.428	0.428
	IQR	−0.0274	0/5	0.002	$0.009^{\downarrow}$
	Isolation Forest	−0.0013	2/5	0.224	0.249
	LOF	−0.0029	0/5	0.085	0.121
	PCA	−0.0017	0/5	0.063	0.104

Table 5: Time-series `lat99` forecasting RMSE: each cleaner vs. NoClean ($n = 25$ replicates).

Probe	Method	ΔRMSE	Wins	p	q
RF	Z-score	−51	10/25	0.179	0.215
	IQR	+21	15/25	0.331	0.361
	Savitzky-Golay	+1199	21/25	0.006	$0.067^{\dagger}$
	Isolation Forest	−257	8/25	0.033	0.078
	LOF	−773	6/25	0.043	0.078
	PCA	+571	16/25	0.026	$0.078^{\dagger}$
kNN	Z-score	−38	11/25	0.022	0.078
	IQR	−33	15/25	0.052	0.078
	Savitzky-Golay	+6	20/25	0.746	0.746
	Isolation Forest	−1385	9/25	0.028	0.078
	LOF	−515	5/25	0.096	0.128
	PCA	−150	11/25	0.050	0.078

Table 6: Time-series `lat99` forecasting MAE: each cleaner vs. NoClean ($n = 25$ replicates).

Probe	Method	ΔMAE	Wins	p	q
RF	Z-score	-44	9/25	0.038	0.057
	IQR	-0.2	12/25	0.978	0.978
	Savitzky-Golay	$+758$	23/25	0.015	$0.057^{\dagger}$
	Isolation Forest	-291	4/25	0.034	0.057
	LOF	-795	7/25	0.026	0.057
	PCA	$+255$	11/25	0.137	0.150
kNN	Z-score	-35	11/25	0.065	0.086
	IQR	-25	14/25	0.034	0.057
	Savitzky-Golay	-115	2/25	0.004	$0.029^{\downarrow}$
	Isolation Forest	-1030	9/25	0.027	0.057
	LOF	-462	4/25	0.109	0.130
	PCA	-152	3/25	0.005	$0.029^{\downarrow}$