Benchmarking Multivariate Time-Series Imputation in 6G Networks

A Comparative Study of Deep Learning and Classical Frameworks

Alessandro Neri[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 21, 2026

Name of the student: Alessandro Neri
Final project course: CSE3000 Research Project
Thesis committee: Rihan Hai, Yuandou Wang, Julian Urbano

Abstract

Sixth-Generation (6G) telecommunications rely on high-frequency millimeter-wave (mmWave) bands for massive data rates, but their physical fragility makes them highly susceptible to line-of-sight blockages. These blockages cause contiguous telemetry outages, creating a single point of failure for edge routing and orchestration protocols demanding continuous system data. To address this, we introduce an evaluation pipeline benchmarking five time-series imputation architectures, from statistical baselines (Nearest Neighbor, Kalman Filter) to complex deep learning models (BRITS, CSDI, TimesNet). Utilizing an open-source microservice dataset, the pipeline dynamically injects simulated blockages across a 24-scenario grid, escalating from minor drops to 60-second outages. Performance is evaluated across an accuracy-latency Pareto frontier. Results demonstrate that the recurrent architecture, BRITS, achieves the highest overall reconstruction fidelity. However, Nearest Neighbor emerges as the optimal low-latency baseline, maintaining competitive accuracy while consistently executing in under 250 milliseconds. Finally, contextualizing these findings reveals a critical limitation: the architectures achieving peak accuracy inherently rely on offline, bidirectional processing to reconcile telemetry gaps. This highlights a significant research opportunity, emphasizing the need to evaluate deep learning models in strictly online, forward-only forecasting configurations to meet the split-second streaming realities of live 6G edge deployment.

1 Introduction

6G is the proposed sixth generation of mobile communications technology, aiming to achieve higher data rates, lower latency, greater energy efficiency, and native AI support. This transition represents a fundamental architectural shift from centralized hardware to distributed, cloud-native Edge Computing. In this system, critical network functions, such as the Access and Mobility Management Function (AMF) and edge-hosted microservices, are deployed as containerized workloads closer to the end-user [10]. To dynamically allocate resources and ensure Quality of Service (QoS), central network orchestrators require continuous streams of system telemetry, including CPU utilization, memory limits, and request latency.

However, the communication links connecting these distributed edge nodes to centralized monitoring systems often utilize high-frequency millimeter-wave (mmWave) or sub-THz bands (30-300 GHz). Although these frequencies provide immense bandwidth, they are highly susceptible to Non-Line-of-Sight (NLoS) blockages from environmental factors such as moving vehicles or urban infrastructure [3]. When a structural blockage severs the link, the central orchestrator experiences a burst of telemetry outage (contiguous gaps in the telemetry). For downstream orchestrators and machine learning applications that rely on continuous system data to make real-time decisions, these data gaps represent a critical point of failure. Furthermore, addressing these gaps in real-time requires *online* (streaming) imputation, which is inherently more difficult than offline processing because algorithms must make immediate estimations without access to future observations [14].

To address this problem, it is crucial to impute the missing edge telemetry as faithfully as possible before it is used to trigger orchestration decisions. Historically, the telecommunications industry has relied on state-space models and classical statistics, such as Kalman Filtering or Nearest Neighbor, to impute this data [9]. More recently, the state-of-the-art has shifted toward complex multivariate deep learning paradigms, including bidirectional recurrent networks (e.g., BRITS [4]), generative diffusion models (e.g., CSDI [11]), and multi-periodic spatial-temporal representations (e.g., TimesNet [12]).

Despite these algorithmic breakthroughs, a knowledge gap remains: these architectures have rarely been subjected to comparative analysis under the specific conditions of a live 6G edge environment. To address this gap, this thesis conducts an evaluation of five imputation architectures (Nearest Neighbor, Kalman Filtering, BRITS, CSDI, and TimesNet) using the EURECOM cloud-native microservices dataset, specifically targeting OpenAirInterface 5G Core AMF system telemetry [10]. By systematically injecting parameterized gaps to simulate real-world physical link blockages, this research aims to answer the following questions:

1. **(RQ1)** How do bursty 6G telemetry outages affect the imputation accuracy (RMSE) across statistical, recurrent, generative, and convolutional models?
2. **(RQ2)** To what extent does the duration and frequency of this contiguous data loss degrade the predictive performance of these imputation architectures?
3. **(RQ3)** What is the Pareto-optimal trade-off between reconstruction accuracy and inference latency when reconciling bursty 6G edge telemetry?

By evaluating these methodologies under simulated real-world conditions, this research delivers a comprehensive evaluation of the accuracy-latency trade-offs. The findings will provide actionable insights for network engineers selecting algorithms for 6G telemetry pipelines.

2 Related Work and Theoretical Background

2.1 Related Work and Gap Analysis

The proliferation of Internet of Things (IoT) devices and distributed edge computing has made continuous time-series data transmission a critical research area. As edge networks scale, they become increasingly susceptible to data loss caused by sensor failures, network latency, and physical communication blockages. Consequently, missing data imputation has been identified as a mandatory component for maintaining reliable and automated orchestration in IoT environments [1].

To address this, recent literature has extensively explored multivariate time-series imputation. The state-of-the-art has rapidly evolved from traditional statistical heuristics to highly complex deep learning paradigms, including Bidirectional Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs). However, as the telecommunications industry shifts toward 6G and cloud-native edge computing, a significant research gap has emerged regarding the practical deployment of these architectures. As noted in recent edge monitoring studies, running imputation models directly at the network edge introduces strict computational and temporal constraints that centralized cloud environments do not face [5].

Specifically, autonomous 6G edge orchestrators, which dynamically manage microservices and route traffic based on incoming telemetry, cannot tolerate the latency of delayed post-processing. They require *online* (streaming) imputation to make immediate routing decisions [14]. Recognizing this constraint, recent research has just begun attempting to adapt deep learning frameworks, such as generative networks, specifically for online edge collaboration to reduce computational overhead [13].

Despite these algorithmic innovations, a knowledge gap remains between theoretical imputation research and practical 6G deployment. Existing literature frequently evaluates time-series architectures under optimal *offline* conditions, focusing only on reconstruction

accuracy while largely ignoring the strict latency and streaming constraints of live network orchestrators. To address this, this study conducts a comprehensive architectural audit. We benchmark a wide spectrum of models (statistical, recurrent, generative, and multi-periodic) to establish their theoretical upper bounds for accuracy, while explicitly measuring their inference latency under simulated bursty outages. We then project these empirical results onto the mathematical and temporal constraints of live 6G edge routing. Ultimately, this research aims to systematically determine whether the complex deep learning configurations currently championed in literature can be practically deployed for real-time 6G orchestration, or if they are inherently restricted to delayed post-processing.

2.2 Formal Problem Description

To utilize interrupted 6G edge telemetry for downstream orchestration tasks, the missing data sequences must be mathematically reconstructed. We formulate this as a multivariate time-series imputation problem, adopting the standard mathematical notation utilized by modern imputation toolkits and recent edge IoT literature [6, 13].

Let the complete, ideal telemetry sequence be defined as a matrix $X \in \mathbb{R}^{T \times D}$, where T is the total number of time steps and D is the number of recorded system features (e.g., CPU usage, memory allocation, and request latency). Because physical network blockages prevent the continuous collection of this telemetry, we introduce a binary masking matrix $M \in \{0,1\}^{T \times D}$, where $M_{t,d} = 1$ if the feature d is successfully observed at time step t, and $M_{t,d} = 0$ if the data point is lost to a connection drop. In this specific edge domain, where network blockages cause bursty telemetry outages, $M_{t,d} = 0$ frequently occurs in long, contiguous sequences across the time dimension T.

The observed dataset X_{obs} can be expressed as the Hadamard product of the true sequence and the mask:

$$X_{obs} = X \odot M$$

The objective of multivariate imputation is to learn a mapping function f_θ that generates a reconstructed matrix $\hat{X}$ based on the available observations:

$$\hat{X} = f_\theta(X_{obs}, M)$$

The optimal parameters θ are obtained by minimizing the error between the reconstructed values $\hat{X}_{t,d}$ and the ground truth $X_{t,d}$ specifically at the coordinate where data was missing ($M_{t,d} = 0$). The methodological framework used to quantify this error is detailed in Section 3.

Furthermore, the deployment of this mapping function introduces a strict temporal constraint. In an *offline* (batched) context, the function f_θ has access to the entire sequence X_{obs}, allowing algorithms to utilize both past and future observations relative to the gap to reconstruct the missing values. However, for live 6G edge orchestrators, the problem becomes an *online* (streaming) task [14]. In this setting, the mapping function must act incrementally, restricted to historical observations up to time t without access to future data. This distinction between online and offline deployment dictates the practical viability of the architectures evaluated in this study.

2.3 Preliminaries of Imputation Methods

To contextualize the evaluation framework, this subsection outlines the theoretical mechanisms that the selected algorithms use to estimate missing time-series data.

Statistical and State-Space Heuristics Traditional imputation methods do not rely on deep neural networks. Heuristics like Nearest Neighbor reconstruct data by identifying and copying the temporally closest available valid observation. State-space models, such as the Kalman Filter, mathematically track the system as a sequence of hidden internal states. They calculate the probability of the next state using linear transition matrices based on past observations. While highly efficient for simple 1D sequences, their reliance on continuous matrix inversions creates computational bottlenecks when applied to highly multivariate datasets [9].

Recurrent Neural Networks (RNNs) Recurrent architectures are explicitly designed to process sequential data by passing hidden memory states from one timestamp to the next. To leverage this in imputation tasks, models such as Bidirectional Recurrent Imputation for Time Series (BRITS) [4] process the sequence both forwards and backwards. This allows the network to interpolate a missing gap by merging the contextual memory from immediately before and after the outage.

Probabilistic Generative Models Unlike deterministic models that predict exact values, probabilistic generative models learn the underlying probability distribution of the dataset. For time-series imputation, architectures like the Conditional Score-based Diffusion Model (CSDI) [11] introduce artificial Gaussian noise into the data and train a neural network to iteratively denoise it. By using the observed data surrounding the gap as a fixed mathematical condition, the model generates plausible reconstructions of the missing segment based on learned historical distributions.

Multi-Periodic Spatial-Temporal Representations Because standard 1D time-series models can struggle to capture complex, overlapping cycles (e.g., daily network traffic rhythms mixed with sudden millisecond latency spikes), multi-periodic architectures (like TimesNet) utilize Fast Fourier Transforms (FFT) [12]. They transform the 1D sequential data into 2D matrices, representing distinct frequency periods, and apply standard 2D Convolutional Neural Networks (CNNs) to extract both intra-period and inter-period correlations simultaneously.

3 Methodology and Pipeline Architecture

This section details the methodology designed to answer the three research questions established in Section 1. We introduce an end-to-end evaluation pipeline that standardizes the processing of multivariate telemetry and simulates physical network blockages. The complete operational workflow of this architecture is visualized in Figure 1. First, Subsection 3.1 outlines the custom gap-injection framework designed to test predictive degradation (RQ2). Next, Subsection 3.2 details the configurations of the imputation architectures selected to benchmark accuracy and latency boundaries (RQ1, RQ3). Finally, Subsection 3.3 establishes the metrics used to evaluate the models' performance.

3.1 Data Preprocessing and Gap-Injection Framework

To ensure standardized evaluation and prevent temporal data leakage, the pipeline ingests raw time-series datasets and dynamically executes an 80/20 chronological train-test split.

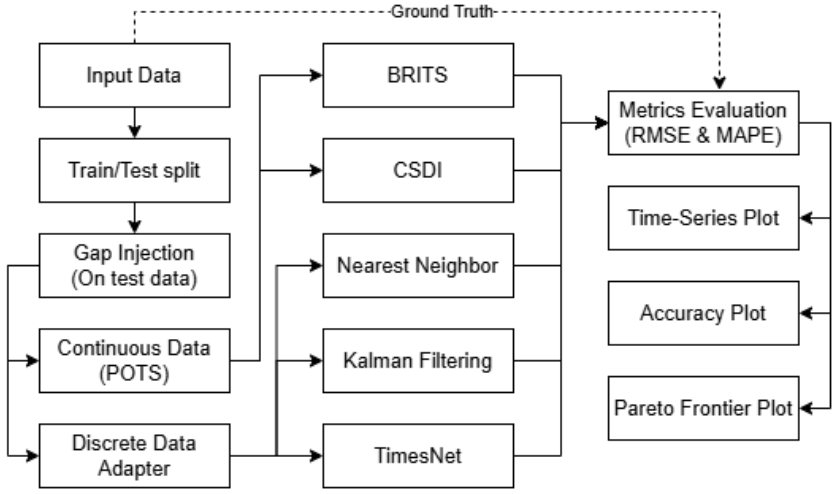


Figure 1: Design of the pipeline, from raw input data to metrics and plots of the performance of the evaluated imputation methods.

The initial 80% of the data is reserved for model training, while the remaining 20% serves as the ground-truth testing sequence. Models requiring equidistant time steps (Kalman Filter, Nearest Neighbor, TimesNet) are discretized into a standardized time grid, while Partially Observed Time-Series (POTS) architectures (BRITS, CSDI) process the continuous timestamps directly.

To explicitly answer RQ2 regarding how contiguous data loss degrades predictive performance, the pipeline uses the 20% ground-truth testing data to automate a 4x6 testing grid. A custom masking function ($M_{t,d} = 0$) injects 24 unique missingness scenarios, pairing four missing data ratios (10%, 25%, 40%, and 50%) with six gap sizes (1, 5, 10, 20, 30, and 60 seconds). This matrix isolates model degradation during prolonged, simulated 6G network outages.

3.2 Selection of Imputation Architectures

To address RQ1 and RQ3, the pipeline benchmarks five models deliberately selected to span the spectrum of computational complexity, establishing clear upper and lower bounds for both accuracy and latency. Because the mathematical theories of these architectures were established in Section 2.3, this subsection strictly details their strategic role within the evaluation framework. To ensure a rigorous and fair comparison, the hyperparameters for the deep learning architectures were systematically tuned using the Optuna optimization framework [2]. This guarantees that all models are evaluated in their optimal, full-context (offline) state to establish their absolute performance ceilings.

To establish the absolute lower bound for computational inference latency (RQ3), we deploy **Nearest Neighbor** in a bidirectional capacity. Alternatively, to represent recursive state-space mathematics, we deploy the **Kalman Filter** [9].

To establish the theoretical upper bound of reconstruction accuracy (RQ1), the pipeline deploys three complex deep learning models. We evaluate sequential memory using **BRITS** [4], utilizing its bidirectional structure to merge past and future telemetry. We deploy **CSDI** [11] to represent the heavy-compute paradigm of probabilistic generative AI, serving as the

latency upper bound for the evaluation. Finally, we integrate **TimesNet** [12] to assess the impact of global multi-periodic spatial-temporal representation on bursty gaps.

3.3 Evaluation Framework

To quantify the performance of the evaluated architectures, the pipeline measures both reconstruction accuracy and computational efficiency.

To answer RQ1, point-wise reconstruction accuracy is evaluated using Root Mean Square Error (RMSE). RMSE is utilized because its squared term heavily penalizes the large prediction errors that disrupt Quality of Service in network routing [7]. To address RQ2, proportional degradation across the 24 scenarios was initially measured using Mean Absolute Percentage Error (MAPE). However, because MAPE divides the absolute error by the true observation, it becomes mathematically unstable when applied to the near-zero baselines frequent in edge telemetry. Consequently, RMSE serves as the primary metric.

Finally, to address RQ3, the pipeline evaluates algorithmic execution speed to establish a Pareto frontier for deployment. In live environments, imputation models must execute within strict latency budgets. To account for this, the pipeline strictly records the forward-pass inference time required by each model to process the missing data blocks. Model initialization and training times are excluded, as real-world edge nodes deploy pre-trained models. By mapping reconstruction error against this computational overhead, the framework identifies the architectures that offer the optimal balance for network deployment.

4 Experiments and Results

4.1 Dataset Description

The primary telemetry utilized in this evaluation is derived from the EURECOM Cloud-Native Microservices dataset, originally published at the IEEE LCN 2022 conference [10]. This repository was specifically designed to benchmark resource provisioning and performance in containerized environments. The multivariate time-series extracted for the imputation matrix (X) consists of critical network health and performance metrics, including CPU utilization, memory limits, memory usage, and end-to-end latency from an OpenAir-Interface 5G Core AMF.

While the initial pipeline validation presented in this study focuses on a single primary dataset, the evaluation framework is structurally designed to ingest and process distinct telemetry datasets from the Zenodo repository. This multi-dataset expansion ensures that the evaluated imputation architectures are tested against diverse network topologies and traffic loads, verifying their generalizability.

4.2 Simulation Environment and Hardware

To ensure the reproducibility of the pipeline and the accurate measurement of algorithmic latency, all experiments were conducted within a standardized hardware and software environment. The evaluation framework was executed on a local workstation equipped with an Intel Core i7-13700H CPU, 16GB of RAM, and an NVIDIA RTX A1000 (6GB) GPU.

The end-to-end evaluation pipeline was orchestrated using Apache Airflow, deployed within an isolated Docker container to guarantee consistent dependency management. The algorithms were implemented in Python; the deep learning architectures (BRITS, CSDI,

TimesNet) were executed using the PyPOTS toolkit [6], while the baseline (Kalman Filter) utilized the Darts library [8]. Lastly, Nearest Neighbor was executed using built-in Pandas methods.

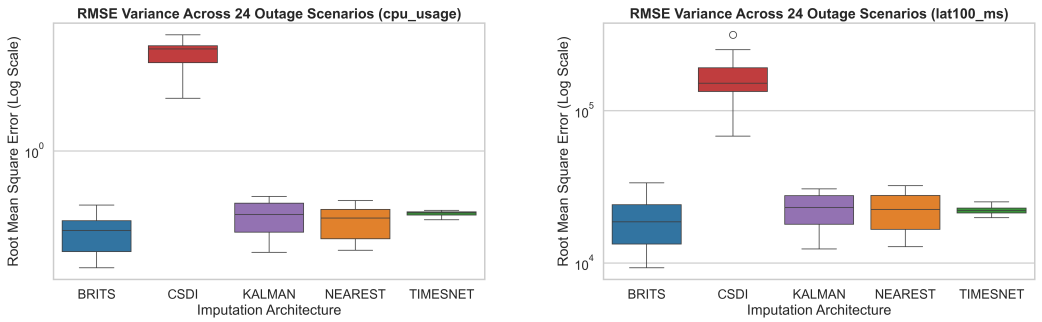
4.3 Quantitative Results

The quantitative results of the 24 dynamic simulation scenarios are evaluated across reconstruction accuracy and algorithmic latency. The aggregate performance of the five imputation architectures, averaged across all missing data ratios and gap durations, is presented in Table 1.

Due to the near-zero baseline characteristic of certain fields (such as fractional CPU cores frequently dropping to small decimal numbers like 0.0001), MAPE exhibits extreme mathematical inflation. Because MAPE calculation divides the absolute error by the true value, a minor deviation from a near-zero baseline results in a percentage error scaling into the millions, rendering MAPE an unreliable comparative metric for certain specific fields. Consequently, RMSE serves as the primary and most mathematically stable metric for evaluating reconstruction accuracy in this environment.

Table 1: Aggregated Summary Table across all gap durations and missing data ratios (AMF 5G Core Telemetry)

Model	Average RMSE	Average MAPE (%)	Average Latency
BRITS	11,652.33	4,867,993.01	1,239 ms (1.2 sec)
Nearest Neighbor	13,267.34	3,910,677.93	99 ms (0.1 sec)
TimesNet	13,381.39	5,709,136.87	12,891 ms (12.9 sec)
Kalman Filter	13,451.34	4,087,074.30	104,711 ms (1.7 min)
CSDI	117,499.78	18,520,517.22	36,154 ms (36.2 sec)



(a) CPU Usage (Fractional Cores)

(b) Max Registration Time (Tail Latency)

Figure 2: Distribution and variance of RMSE (Log Scale) across all 24 simulated 6G outage configurations (AMF 5G Core Telemetry dataset).

Reconstruction Accuracy (RMSE) An analysis of the RMSE distribution across the 24 unique simulation scenarios reveals distinct performance tiers and variance profiles among the evaluated models. Represented on a logarithmic scale, these box plots (Figures 2a and

2b) illustrate the operational stability of the architectures under escalating gap sizes and missing data ratios.

The recurrent deep learning architecture, BRITS, recorded the lowest aggregate RMSE overall (11,652.33). While it exhibits moderate variance across the different outage scenarios, its entire error distribution remains firmly at the lower bound of the scale, successfully minimizing extreme outliers. It achieved an average RMSE of 0.28 for CPU usage, and 19,233.13 ms for maximum registration latency.

Interestingly, the multi-periodic architecture, TimesNet, demonstrated the tightest error variance across all metrics. This indicates that TimesNet's reconstruction stability is very resilient to varying gap sizes, even though its average error is slightly higher than that of BRITS and Nearest Neighbor.

The statistical baseline, Nearest Neighbor, exhibited the second-lowest aggregate error (13,267.34), maintaining a variance profile closely resembling the Kalman Filter (13,451.34) but consistently outperforming it. Conversely, the generative diffusion model, CSDI, yielded the highest point-wise errors. Furthermore, it exhibited massive predictive instability in CPU usage and maximum registration latency, placing its distribution an order of magnitude higher than the other evaluated architectures.

Latency and Pareto Frontier The distribution of the models across the accuracy-latency trade-off space is visualized via Pareto scatter plots (e.g., Figure 3), which map the average RMSE against the inference latency on a logarithmic scale.

Nearest Neighbor recorded the lowest average inference latency at 99 milliseconds. Remarkably, the recurrent deep learning architecture, BRITS, demonstrated highly efficient execution, averaging just 1,239 milliseconds (1.2 seconds) per inference block. TimesNet and CSDI required 12,891 milliseconds (12.9 seconds) and 36,154 milliseconds (36.2 seconds), respectively. Conversely, the recursive statistical baseline, Kalman Filter, proved to be the most computationally expensive during inference, averaging 104,711 milliseconds (1.7 minutes). This highlights the architectural difference between highly parallelized deep learning tensor operations and strictly chronological, sequential state-space updates. An evaluation of the reconstructed time-series waveforms during these blockages is available in Appendix C, providing further visual confirmation of the models' point-wise behaviors.

Predictive Degradation under Prolonged Outages (RQ2) To evaluate how contiguous data loss impacts predictive performance, reconstruction error was mapped across the escalating gap sizes (Figure 4). The empirical results contradict the assumption that predictive degradation scales linearly with outage duration. Instead, BRITS, Nearest Neighbor, and the Kalman Filter exhibit an initial error increase at smaller gaps (1 to 20 seconds) before stabilizing at larger durations. Notably, TimesNet maintains a remarkably flat error profile across all gap sizes, mathematically validating the tight variance observed in Figure 2a. Conversely, CSDI maintains a consistently massive error magnitude regardless of gap duration, highlighting that its failure stems from an inability to map the physical constraints of the telemetry rather than a sensitivity to gap size.

Cross-Domain Generalization (Python Web Server Telemetry) To ensure the evaluated architectural trade-offs are fundamental to the imputation models rather than specific anomalies of the 5G AMF dataset, the evaluation pipeline was additionally executed against a secondary domain: application-layer telemetry from a Python Web Server

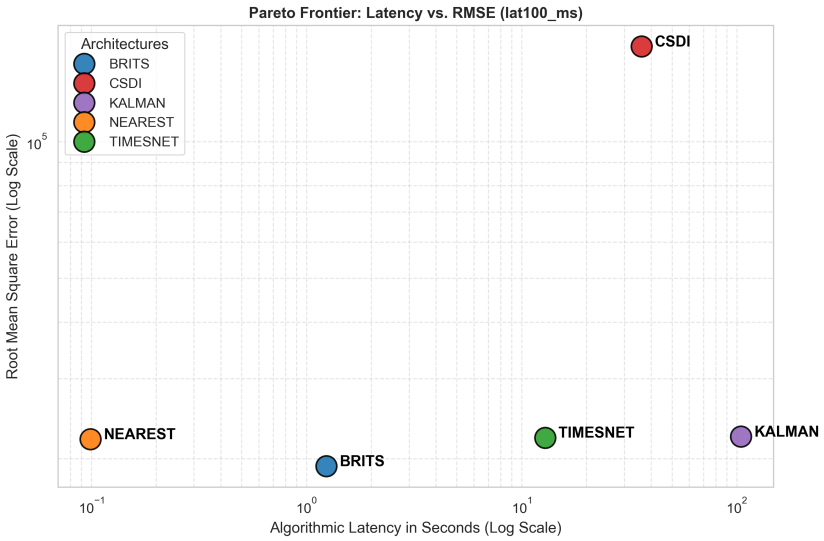


Figure 3: Pareto frontier plotting the average reconstruction accuracy (RMSE) of Maximum Registration Time (lat100_ms) against inference latency (Log Scale).

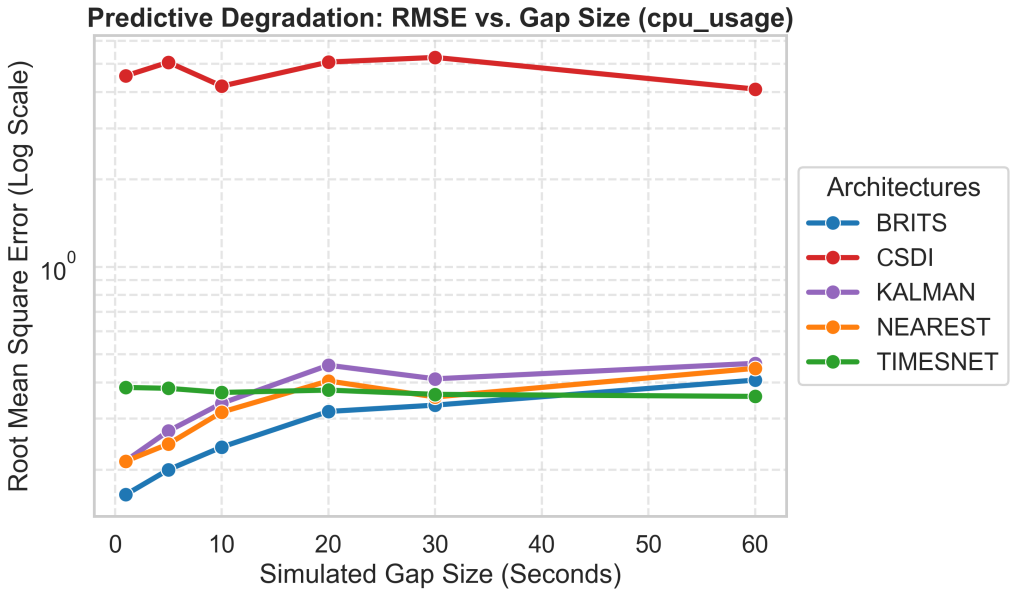


Figure 4: Inference accuracy degradation (RMSE) across increasing simulated gap sizes for CPU Usage.

(Table 2). The empirical results across this secondary dataset demonstrated a near-perfect replication of the patterns and results observed in the AMF environment.

BRITS maintained its position as the most accurate model (RMSE: 8,031.73). However, its average inference latency scaled to 10.6 seconds in this domain, reflecting the sensitivity of recurrent architectures to varying sequence lengths and dataset dimensionalities. Consequently, Nearest Neighbor strongly retained its status as the Pareto-optimal model, achieving highly competitive accuracy (RMSE: 8,172.20) at a fraction of the computational cost (232 milliseconds). Furthermore, the generative diffusion architecture (CSDI) continued to exhibit low accuracy, yielding the highest point-wise error (RMSE: 35,879.94). Finally, the statistical baseline paradox remained consistent; the recursive, step-by-step nature of the Kalman Filter resulted in the highest inference latency (1.7 minutes) across both domains. This cross-domain consistency confirms that the accuracy and latency identified in this study generalize across different layers of edge network telemetry.

Table 2: Aggregated Summary Table across all gap durations and missing data ratios (Python Web Server Telemetry)

Model	Average RMSE	Average MAPE (%)	Average Latency
BRITS	8,031.73	31,994.81	10,618 ms (10.6 sec)
Nearest Neighbor	8,172.20	25,109.83	232 ms (0.2 sec)
TimesNet	11,023.49	74,141.63	19,281 ms (19.3 sec)
Kalman Filter	31,462.42	45,021.39	104,510 ms (1.7 min)
CSDI	35,879.94	58,598.80	35,677 ms (35.7 sec)

5 Responsible Research

5.1 Reproducibility and Pipeline Orchestration

To ensure the evaluation framework is reproducible, the end-to-end pipeline was orchestrated using Apache Airflow and deployed within an isolated Docker container, guaranteeing consistent software dependency management across different operating systems. The Directed Acyclic Graph (DAG) managing the architectural evaluation and metric generation is visually documented in Figure 5, providing evidence of the automated orchestration logic. To further guarantee deterministic execution, the pseudo-random number generator utilized in the gap-injection framework, the only stochastic component during data preparation, is hardcoded to a fixed initialization state (`np.random.seed(42)`).

The complete evaluation framework, including the Python preprocessing scripts, imputation algorithms, and the specific telemetry dataset subset utilized for validation, has been publicly released under the open-source Apache-2.0 license, alongside an interactive Streamlit dashboard (Appendix B) at `https://github.com/neriAle/6G-Time-Series-Imputation`. By open-sourcing this deterministic pipeline, independent researchers can recreate the exact 24 experimental scenarios using the standard hardware parameters detailed in Subsection 4.2.

5.2 Ethical Considerations

This research was conducted in strict adherence to institutional guidelines regarding data privacy, network safety, and responsible computational practices.

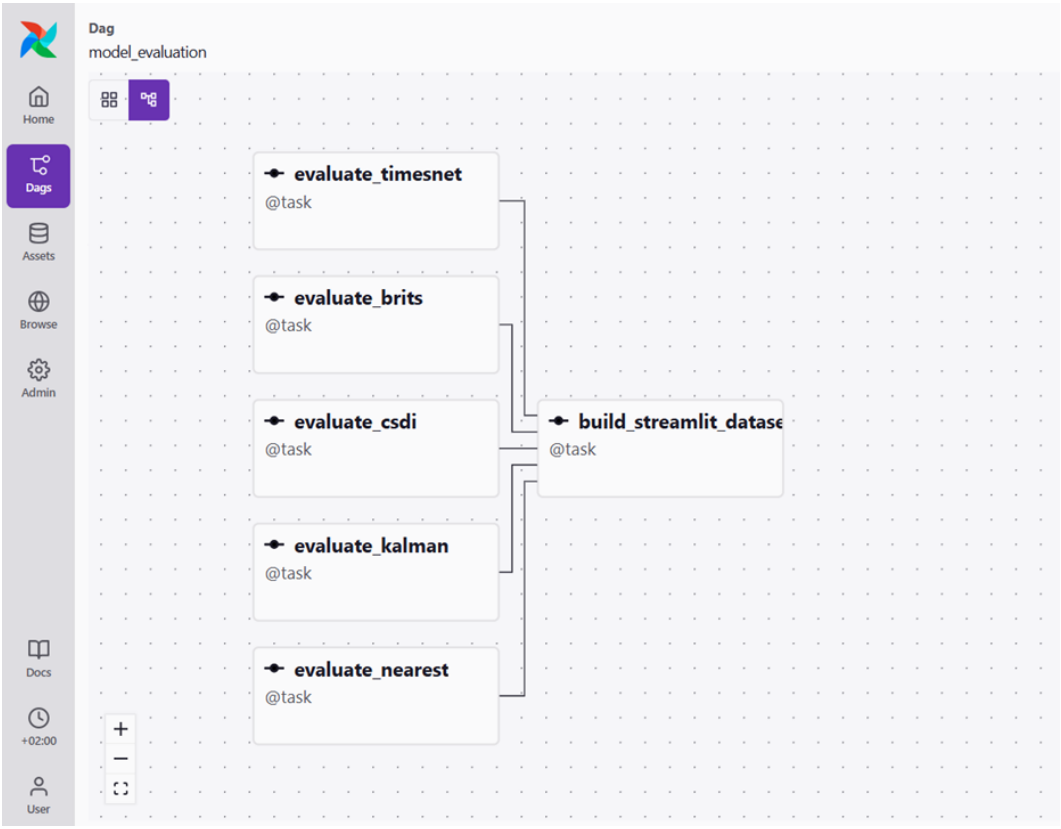


Figure 5: Apache Airflow Directed Acyclic Graph (DAG) demonstrating the automated task dependencies, where parallel architecture evaluations converge into a unified dataset builder.

Data Privacy The telemetry utilized in this study, sourced from a public Zenodo repository designed for microservice benchmarking [10], consists entirely of network performance metrics (e.g. CPU utilization, memory limits, and signal latency). It contains zero Personally Identifiable Information (PII), completely eliminating risks related to data privacy and surveillance.

Reliability and Downstream Risks In automated 6G environments, the consequences of algorithmic inaccuracy extend beyond mathematical error metrics. If an imputation architecture confidently generates incorrect telemetry, such as hallucinating a low-latency state during a severe physical blockage, it could mislead downstream routing protocols into forwarding traffic directly into a network dead zone. This false confidence presents a critical reliability risk. Consequently, the ethical deployment of these algorithms requires evaluating not just their average accuracy but also their worst-case boundary failures to prevent automated cascading outages.

Environmental Impact It is important to acknowledge the environmental and computational overhead associated with modern imputation architectures. Empirical observations

from this study reveal that generative and multi-periodic deep learning models consume immense computational resources. Probabilistic models such as CSDI require heavy iteration for reverse-denoising, while architectures like TimesNet introduce massive overhead through continuous Fast Fourier Transforms (FFT) and 2D tensor convolutions [12]. Similarly, the classical state-space Kalman Filter, with its strictly sequential nature and $O(d^3)$ matrix inversion complexity for multivariate data, leads to severe CPU execution bottlenecks. As 6G networks scale, deploying mathematically intensive architectures at the edge must be critically weighed against their carbon footprint, hardware dependencies, and power consumption constraints.

6 Discussion

Evaluating time-series imputation under simulated 6G blockages reveals a significant gap between theoretical model performance and real-world edge deployment. This section interprets these findings, detailing the accuracy-latency trade-offs and identifying the architectural mechanisms driving model success or failure.

6.1 Balancing Accuracy and Speed at the 6G Edge

In 6G edge networks, execution speed is as critical as reconstruction fidelity. The Pareto frontier (Figure 3) highlights a stark divide between the operational practicality of simple heuristics and the computational overhead of complex deep learning.

Nearest Neighbor emerges as the most practical choice for immediate edge deployment, executing in under 250 milliseconds across both domains. Its bidirectional configuration leverages the inherent short-term stability of network telemetry (e.g., CPU allocation), pulling the closest valid observation to achieve highly effective imputation well within the strict latency budgets of microservice orchestrators.

Conversely, capturing complex, non-linear shifts (e.g., latency spikes) requires deep learning architectures like BRITS. BRITS achieved the lowest overall error with highly efficient inference (1.2 seconds in the AMF domain). However, its reliance on offline, bidirectional processing necessitates waiting for future context. This renders it highly effective for delayed analytics but structurally constrained for immediate, split-second online routing.

6.2 Why the Models Succeeded or Failed

An analysis of the empirical results reveals how the mathematical designs of these architectures clash with the physical constraints of live network data.

CSDI Error Analysis The generative diffusion model, CSDI, yielded the highest errors by essentially "hallucinating" data (Appendix C). Designed to generate continuous media like audio or images, CSDI lacks inherent constraints for physical hardware limits. Consequently, it predicted impossible network telemetry, such as negative latency or exceeding maximum CPU cores, causing its error rate to skyrocket when deprived of physical boundaries.

Sample Efficiency Constraints CSDI's severe hallucination also highlights a critical disparity in sample efficiency. While recurrent architectures like BRITS successfully learned telemetry dynamics from the 80% training split, generative diffusion notoriously requires

exponentially larger datasets to map probability spaces. This massive data dependency renders CSDI computationally impractical for lightweight, dynamic 6G edge environments.

Kalman Filter Latency Characteristics Empirical tests confirmed the Kalman Filter's severe computational bottleneck, proving it the slowest evaluated architecture (averaging 1.7 minutes per inference). This latency stems directly from its $O(d^3)$ matrix inversion complexity and strict sequential updating. While deep learning utilizes highly parallelized tensor operations, classical step-by-step recursive updates severely hinder real-time multivariate prediction.

BRITS Performance Analysis BRITS succeeded primarily due to its bidirectional configuration. While forward-only streaming models lose system state during connection drops, BRITS processes context from both before and after the outage. Merging these perspectives allows it to accurately bridge gaps and catch sudden latency spikes that monodirectional methods miss.

Discretization Constraints versus POTS Accuracy divergence heavily depends on handling native temporal sparsity. Real-world 6G logs often contain native gaps of 10 to 60 seconds. Architectures requiring equidistant time steps (TimesNet, Kalman Filter, Nearest Neighbor) force this data into a rigid 1-second grid. Consequently, minor blockages inside large native gaps force massive computational over-imputation (e.g., imputing 120 artificial points for a 2-second gap). While this universally increases inference latency for these models, the impact on predictive accuracy diverges significantly. As shown in Figure 4, the sequential Kalman Filter suffers an initial inflated error spike before eventually flattening out. TimesNet, however, absorbs this discretization burden without predictive degradation; its multi-periodic architecture maintains a flat, resilient error profile regardless of gap size. Conversely, POTS architectures (BRITS, CSDI) process continuous timestamps directly. By calculating only the exact missing values, they completely bypass both the latency and error penalties of the "discretization penalty" in irregular network environments.

7 Conclusions and Future Work

This study addressed the challenge of bursty data loss in emerging 6G mmWave networks by designing an evaluation pipeline. We benchmarked five time-series imputation architectures, ranging from traditional statistical methods to modern deep learning models, across two distinct edge telemetry datasets. Our primary objective was to evaluate these architectures against three research questions regarding accuracy, degradation under prolonged data loss, and operational latency constraints.

7.1 Answers to Research Questions

(RQ1) *How do bursty 6G telemetry outages affect the imputation accuracy (RMSE) across statistical, recurrent, generative, and convolutional models?*

The results demonstrate that BRITS provides the highest reconstruction accuracy (lowest RMSE) during outages. It successfully reconstructs complex, non-linear patterns, such as isolated and sudden spikes, by bridging hidden memory states from both directions of

the gap. However, while the traditional statistical baseline (Nearest Neighbor) maintains competitive accuracy, the generative diffusion architecture (CSDI) fails entirely in this domain. CSDI's lack of awareness regarding the physical boundaries of infrastructure telemetry resulted in severe data hallucination and exponential errors.

(RQ2) *To what extent does the duration and frequency of this contiguous data loss degrade the predictive performance of these imputation architectures?*

Contrary to expectations, predictive degradation does not scale linearly with the size or frequency of contiguous data loss. Instead, degradation is highly dependent on the mathematical nature of the specific telemetry feature and the sampling frequency of the dataset. Because many edge telemetry metrics (e.g., CPU usage) exhibit long periods of baseline stability only interrupted by brief sparse spikes, wider simulated gaps frequently overlap with constant data periods. Consequently, the error curves for the most accurate architectures (BRITS, Nearest Neighbor) remain stable during prolonged outages.

Furthermore, as demonstrated in Figure 4, architectures requiring equidistant discretization experience highly divergent outcomes as gaps widen. While the multi-periodic TimesNet proves remarkably resilient to gap size, maintaining a consistently flat error profile, the sequential Kalman Filter suffers a severe initial error spike before leveling off into a high foundational baseline. In contrast, POTS architectures (like BRITS) dynamically adapt to the continuous timestamps directly, completely bypassing the penalties of rigid discretization while achieving superior overall accuracy.

(RQ3) *What is the Pareto-optimal trade-off between reconstruction accuracy and inference latency when reconciling bursty 6G edge telemetry?*

The Pareto-optimal trade-off depends entirely on the specific operational requirement of the network monitoring system. If the primary goal is ultra-low latency, **Nearest Neighbor** is the optimal method, consistently executing in under 250 milliseconds while providing sufficient accuracy for stable baseline metrics. However, if maximum reconstruction fidelity is required to map complex spikes, **BRITS** represents the optimal deep learning compromise. By isolating the inference phase from the training overhead, we demonstrated that the pre-trained recurrent model can achieve the highest overall accuracy while maintaining a highly efficient inference speed of approximately 1.2 seconds within the primary 5G AMF environment. Conversely, the remaining evaluated architectures (Kalman Filter, TimesNet, and CSDI) are strictly non-optimal, suffering from prohibitive computational bottlenecks or severe discretization penalties.

7.2 Limitations of the Current Study

While the evaluation pipeline provides a robust framework for comparing imputation architectures, we acknowledge three primary methodological limitations:

- **Isolated Inference versus System-Level Latency:** The current pipeline evaluates inference latency within a controlled, isolated software environment. Transitioning these models to active 6G edge microservices may introduce system-level orchestration delays and networking overheads that were not observed in this study.
- **Online Streaming versus Offline Batched Deployment Constraints:** The architectures that achieved the highest accuracy in this study (BRITS and bidirectional

Nearest Neighbor) utilized offline, batched processing to merge past and future context to reconcile the telemetry gaps. However, for strictly *live* 6G edge routing tasks, algorithms must make immediate decisions on streaming data incrementally, without access to future observations [14]. While the evaluated architectures natively support online forecasting configurations, evaluating them in a forward-only capacity fell outside the scope of this baseline study. Stripping these models of future context will inherently alter their accuracy profiles, making their evaluation in strictly online scenarios a necessary next step.

- **Artificial Gaps:** Although the gap-injection framework accurately simulates the contiguous bursty nature of physical NLoS blockages, these gaps were artificially induced via boolean masking ($M_{t,d} = 0$). While this approach is sufficient for baseline evaluation, physical hardware disconnections may trigger additional complex network behaviors that are not fully represented by our masking approach.

7.3 Future Work and Recommendations

To address these open issues, we formulate two explicit recommendations for future network engineering research. First, subsequent iterations of this pipeline should be deployed directly onto actual resource-constrained edge hardware, such as Raspberry Pi clusters. This will allow researchers to measure the true system-level operational latency of deep learning inference. Second, future algorithmic research must focus exclusively on evaluating these architectures in their *online, forward-only* forecasting configurations. By measuring how predictive accuracy degrades when bidirectional context is removed, researchers can definitively determine if deep learning models can provide tensor-accelerated speeds and high accuracy while strictly adhering to the streaming realities of live, split-second 6G edge routing.

References

[1] Deepak Adhikari, Wei Jiang, Jinyu Zhan, Danda B Rawat, Uwe Aickelin, and Hadi Khorshidi. A comprehensive survey on imputation of missing data in internet of things. *ACM Computing Surveys*, 55, 05 2022.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Ahmed Alkhateeb, Gouranga Charan, Tawfik Osman, Andrew Hredzak, Joao Morais, Umut Demirhan, and Nikhil Srinivas. Deepsense 6g: A large-scale real-world multi-modal sensing and communication dataset. *IEEE Communications Magazine*, 61(9):122–128, 2023.

[4] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series, 2018.

[5] Alexandru Cioca. *Missing value imputation methods at edge IoT monitoring networks*. Projecte final de màster oficial, UPC, Facultat d'Informàtica de Barcelona, Departament d'Arquitectura de Computadors, 07 2025.

[6] Wenjie Du, Yiyuan Yang, Linglong Qian, Jun Wang, and Qingsong Wen. Pypots: A python toolkit for machine learning on partially-observed time series. *arXiv preprint arXiv:2305.18811*, 2023.

[7] GoML. Time series forecasting: From basics to advanced predictive analysis, Jan 2024. Accessed: 2026-05-22.

[8] Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan Kościsz, Dennis Bader, Frédérick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and Gaël Grosch. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022.

[9] Rudolf E. Kalman. A new approach to linear filtering and prediction problems, 1960.

[10] M. Mekki, N. Toumi, and A. Ksentini. Benchmarking on Microservices Configurations and the Impact on the Performance in Cloud Native Environments, July 2022.

[11] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation, 2021.

[12] Haixu Wu, Tengge Hu, Yong Liu, Hang Dong, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[13] Zhaokang Zhan, Dazhong Ma, Xuguang Hu, and Siqi Zhang. An online collaborative imputation method for industrial missing data based on multiscale matgan in edge computing. *IEEE Internet of Things Journal*, 12(10):14244–14253, 2025.

[14] Yuxuan Zhao, Eric Landgrebe, Eliot Shekhtman, and Madeleine Udell. Online missing value imputation and correlation change detection for mixed-type data via gaussian copula. *CoRR*, abs/2009.12326, 2020.

A Appendix A: Generative AI Disclosure

In accordance with TU Delft academic integrity guidelines, the author discloses the use of a generative AI assistant (Google Gemini 3.1 Pro) during the preparation of this manuscript. The tool was utilized strictly as an assistive software to support the research process. Its applications were limited to brainstorming methodological structures, enhancing the fluency and clarity of academic prose, and providing LaTeX formatting assistance. The AI was not used to generate core research ideas, analyze results, or write code for the evaluation pipeline. The author retains full creative, intellectual, and academic responsibility for the design of the experiments, the accuracy of the physical mmWave mechanics described, and the final conclusions presented in this study.

B Appendix B: Streamlit Evaluation Dashboard

To facilitate the interactive exploration of the reconstructed telemetry and error distributions, a custom Streamlit dashboard was developed as part of this thesis. Figure 6 provides

a visual overview of the user interface, which allows researchers to dynamically filter the 24 outage scenarios and instantly visualize the performance of the imputation architectures and their trade-offs. Figure 7 shows the second page of the Streamlit dashboard: the Time-Series reconstruction, a useful way to visualize the ground truth, the masked values, and the values reconstructed by the five evaluated imputation architectures.

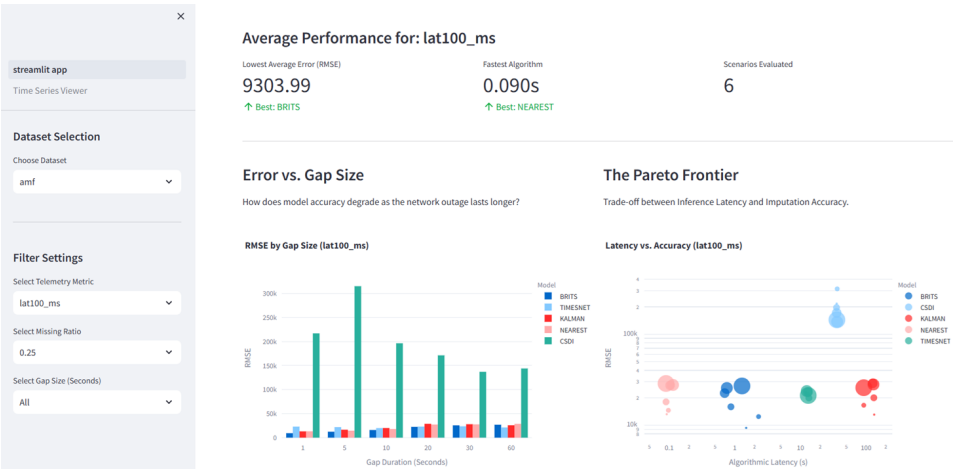


Figure 6: User interface of the custom Streamlit dashboard developed to interactively evaluate the imputation pipeline results.

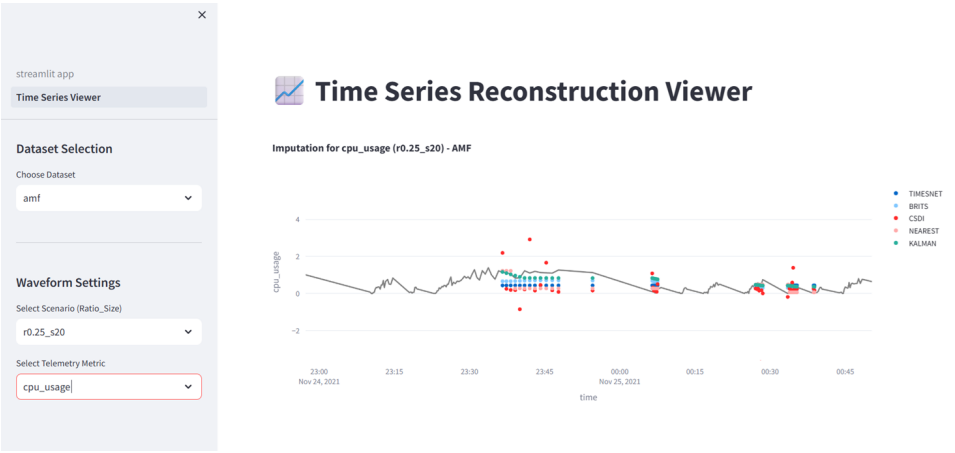


Figure 7: Streamlit dashboard page to visualize the ground truth and imputed values, filterable by dataset, column and scenario.

C Appendix C: Supplementary Waveform Reconstructions

To provide qualitative visual confirmation of the models' point-wise behaviors inside the induced gaps, Figure 8 illustrates a time-series waveform reconstruction during a simulated 6G telemetry outage. As discussed in Section 6, this visualization highlights the ability of bidirectional processing (BRITS) to catch sudden, non-linear latency spikes, while also demonstrating the severe hallucination effect exhibited by generative models (CSDI) when deprived of physical hardware boundaries.

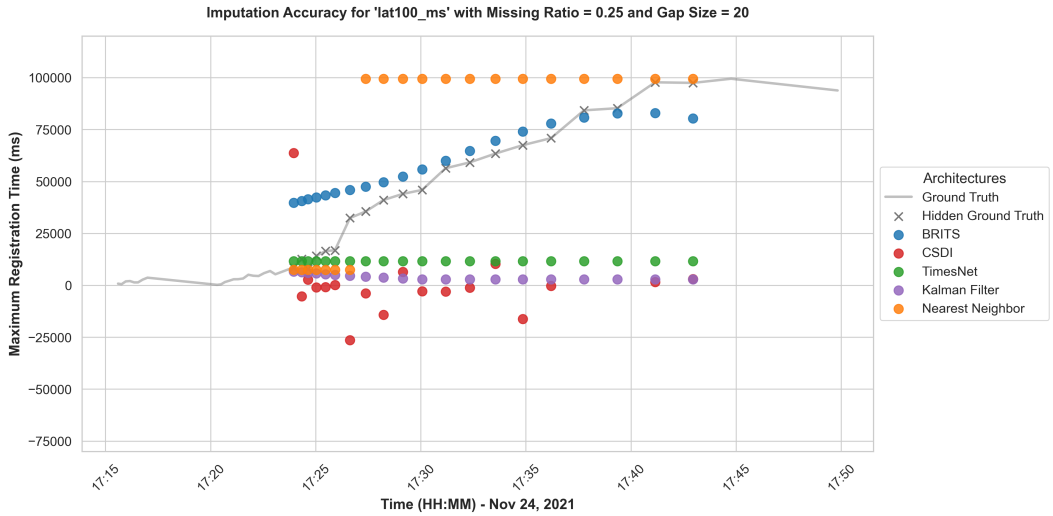


Figure 8: Time-series waveform reconstruction of Maximum Registration Time (lat100_ms) during a 20-second simulated blockage (with a missing ratio of 0.25), overlaying multi-architecture predictions directly over the masked ground truth indicators.