Simulated Annealing-Based Imputation for Large Gaps in Big Science Network Time Series

Sanjay Chari
Rensselaer Polytechnic Institute
Troy, New York, USA
charis3@rpi.edu

Kevin A. Brown
Argonne National Laboratory
Argonne, Illinois, USA
kabrown@anl.gov

Andrew Norman
Fermi National Accelerator Laboratory
Batavia, Illinois, USA
anorman@fnal.gov

Christopher D. Carothers
Rensselaer Polytechnic Institute
Troy, New York, USA
chrisc@cs.rpi.edu

ABSTRACT

Long, uninterrupted gaps in time series data can significantly hinder downstream analysis, particularly in scientific and operational environments. Standard imputation techniques, such as forward or backward filling, often distort temporal structure or violate known constraints, resulting in unrealistic values and degraded predictive performance. This paper introduces a simulated annealing-based approach designed to address these challenges. Our method frames imputation as a constrained optimization problem that balances smoothness, periodic trends, and monotonic behavior. We evaluate our framework on real-world network telemetry from the Energy Sciences Network (ESNet), specifically the connection between Fermilab and Argonne National Laboratory. Results show that our method (GAPSA) produces realistic imputations that align with expected traffic patterns and capture the variance in the data better while achieving comparable MSE and SMAPE values as state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies**; • **Networks**; • **Information systems**; • **Computer systems organization**;

KEYWORDS

Data interpolation, Data mining, Time series analysis, Irregular time series, Time series imputation, Time series forecasting, Simulated annealing

ACM Reference Format:
Sanjay Chari, Kevin A. Brown, Andrew Norman, and Christopher D. Carothers. 2025. Simulated Annealing-Based Imputation for Large Gaps in Big Science Network Time Series. In *Proceedings of KDD MILETS Workshop 2025*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD MILETS Workshop 2025, August 4, 2025, Toronto, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This study is part of the Tachyon Project [4], which explores new approaches to modeling the infrastructure involved in large-scale scientific computing workflows associated with the Deep Underground Neutrino Experiment (DUNE) [1]. DUNE generates vast quantities of data from liquid argon detectors at Fermilab (FNAL), which are transferred for data analysis to supercomputers at Argonne National Laboratory (ANL) via the Energy Sciences Network (ESNet). As a result, ESNet becomes a critical component of the experimental pipeline, and understanding its behavior is an integral part of the project.

ESNet telemetry offers a window into the behavior of data transfers between FNAL and Argonne. It logs metrics like byte counters and packet counts at regular intervals, which can be used to monitor performance, detect anomalies, and predict network usage. But these insights rely on having continuous data. In reality, equipment maintenance, outages, or logging errors often create large gaps. In our case, over four months of data are missing due to corrupted database issues.

Filling these kinds of gaps is far from trivial. Common methods like forward fill or linear interpolation are easy to apply, but they can flatten important trends or introduce sharp transitions that don't reflect how the system actually behaved. More sophisticated models like BRITS [2] or SAITS [5] use machine learning to estimate missing values, but they usually expect scattered missing points. When faced with an extended blackout, they tend to fall back on overly smooth or generic estimates that fail to capture real dynamics.

Our approach takes a different route. We treat imputation as a constrained optimization problem and solve it by minimizing a quadratic objective using simulated annealing (SA). We name our approach GAPSA (Gap-Aware Probabilistic Simulated Annealing). The problem is formulated as a QUBO (Quadratic Unconstrained Binary Optimization), where each possible value for a missing time step is mapped to a binary variable, and the overall configuration is scored based on how well it aligns with expected patterns. The objective function penalizes deviations from a smooth trend, violations of typical value ranges, and sharp transitions across adjacent time steps. It also includes soft priors when available, nudging the solution toward plausible domain-informed patterns. A one-hot constraint ensures that exactly one value is chosen per time step. We perform simulated annealing over this QUBO, guiding the search

through the space of discrete imputations. This setup ensures that the imputed segment not only fits known temporal structure but also satisfies hard endpoint constraints, such as matching the total increase across the gap.

To evaluate the method, we apply it to two different datasets: telemetry from the FNAL–ANL ESNet link, and a benchmark air quality dataset from UCI [22]. This allows us to test both domain-specific and general-purpose imputation. Across both cases, we compare our method against traditional and deep learning-based techniques, not just by reconstruction error, but by how well they support forecasting models trained on the filled data. Our results show that the simulated annealing approach holds up well, often producing more coherent and interpretable fills. It reflects the real-world patterns better than generic smoothers, which helps both analysts and predictive models make sense of the data.

2 RELATED WORK

Time series imputation has been widely studied, with early methods relying on statistical heuristics such as forward fill, backward fill, linear interpolation, or exponential smoothing [7]. While these methods are computationally cheap, they tend to distort seasonal patterns and underperform in the presence of long or structured gaps.

More sophisticated approaches apply statistical models, such as Kalman smoothing or Expectation-Maximization (EM) techniques, which attempt to infer missing values using learned temporal dynamics [6]. However, these models assume certain distributions or stationarity, and their accuracy degrades with large contiguous gaps. Low-rank matrix completion methods [23] and nearest-neighbor-based strategies like MissForest and kNN imputation [19] offer improvements but often struggle with temporal coherence and global structure.

Deep learning has recently emerged as a dominant paradigm for imputation. Models like BRITS [2] and GRU-D [3] use recurrent neural networks to iteratively refine missing values, treating them as learnable parameters. Transformer-based models such as SAITS [5] and PatchTST [15] incorporate attention mechanisms to exploit long-range dependencies across both time and features. These methods perform well when trained on large, representative datasets, but often assume that missingness is random or scattered, which limits their ability to handle large, contiguous gaps without retraining.

Simulated annealing (SA) has been used in various optimization contexts for time series, including imputation, but is typically limited to simpler settings such as short gaps or single-variable sequences. For example, works like [14] used SA for small-scale interpolation problems or outlier smoothing. However, they do not model the gap as a structured constrained optimization problem, nor do they integrate prior knowledge about traffic behavior or domain constraints.

Our approach differs by formulating imputation as a discrete quadratic unconstrained binary optimization (QUBO) problem over plausible values for each time point in the gap. The energy function integrates multiple terms: fidelity to a seasonal prior, smoothness relative to expected slope, range constraints, and hard endpoint consistency. A one-hot encoding ensures that only a single candidate value is selected per time point. The QUBO is then solved using classical simulated annealing. This makes our method both interpretable and easily tunable, while allowing hard constraints and priors to be encoded directly into the optimization. Compared to neural models, our approach requires no retraining, adapts to each individual gap, and can incorporate domain knowledge (e.g., from similar past intervals or known traffic profiles) via a soft prior term.

3 EXPLORATORY DATA ANALYSIS

3.1 Data Wrangling

The raw network data contained information about the number of broadcast, unicast, multicast, and cumulative number of packets coming in and going out of each interface of the ESNet router at the respective site. Since we were interested in the larger patterns of network traffic at each site, we summed up the data across all interfaces for a router at a given point in time.

There were a number of challenges with handling the time series data. First, the data had irregular intervals. The intervals varied randomly within a range of an hour, from 5 minutes to 35 minutes between consecutive data points. Additionally, the dataset had data points for September 2024 and February 2025 but no data points in between, causing a large gap. To address these issues, we first applied backward fill to regularize the data to intervals of 10 minutes and then imputed the gap with various methods - Forward Fill, Backfill, Exponential Smoothing, SAITS, BRITS, and our method GAPSA.

3.2 Feature Description and Selection

The router telemetry dataset used in this study contains several features that reflect different aspects of network traffic. These include counters for incoming and outgoing multicast and broadcast packets, high-capacity octet transmission, and various other packet-level metrics. Each feature represents a specific measurement collected at regular intervals from network routers, offering a detailed view of how traffic flows through the network.

Among these features, we focus on two in particular: ifHCOutBroadcastPkts and ifHCOutOctets. The first captures the number of broadcast packets transmitted by the router, while the second records the total number of octets (bytes) sent out. Together, they serve as strong indicators of network utilization and traffic volume.

Our decision to focus on these two features is guided by the objectives of the Tachyon project. One of the project's key goals is to develop robust models that can accurately forecast bandwidth demands within the ESNet infrastructure. This capability is crucial for optimizing data transfer pipelines supporting large-scale scientific experiments like DUNE. By modeling and forecasting these metrics, we aim to better understand usage patterns and anticipate periods of high load or potential congestion. These insights can, in turn, inform scheduling, resource allocation, and anomaly detection strategies across the scientific computing workflow.

3.3 Observed Weekday Trend Patterns

We examine average network activity patterns by day of the week using ESNet telemetry data between Fermilab and Argonne National Laboratory. Aggregating the data by weekday reveals characteristic behaviors that vary across features, helping identify usage cycles that may align with workweek dynamics or scheduled tasks.

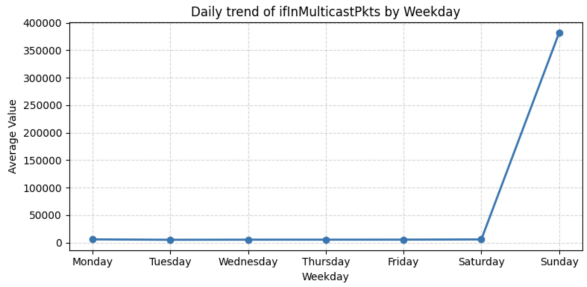


Figure 1: Weekday trend of `ifInMulticastPkts`. The x-axis shows weekdays, and the y-axis indicates the average incoming multicast packet count. The trend shows minimal activity throughout the week, followed by a sharp and dramatic peak on Sunday.

1. Dramatic Sunday Peak: Certain features, notably `ifInMulticastPkts`, exhibit low average values consistently from Monday to Saturday, followed by an abrupt and significant peak on Sunday (Figure 1). This suggests scheduled multicast-heavy operations that specifically occur at week's end.

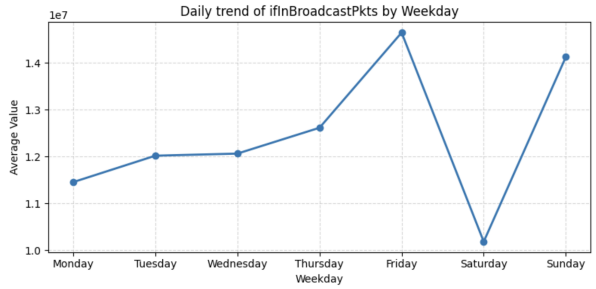


Figure 2: Weekday trend of `ifInBroadcastPkts`. The x-axis shows weekdays, and the y-axis represents average incoming broadcast packet count. Activity generally rises through the weekdays, peaks significantly on Friday, sharply drops on Saturday, and partially recovers on Sunday.

2. End-of-Week Peak with Weekend Drop: In cases such as `ifInBroadcastPkts`, network activity progressively increases from Monday, reaching a pronounced peak on Friday, then sharply declining on Saturday before a partial recovery on Sunday (Figure 2). This could indicate high usage patterns related to weekly business or batch processes concluding before the weekend.

3. Multiple Midweek and Weekend Peaks: Features like `ifOutBroadcastPkts` present oscillating weekly behavior characterized by midweek and weekend peaks (Figure 3). The pattern

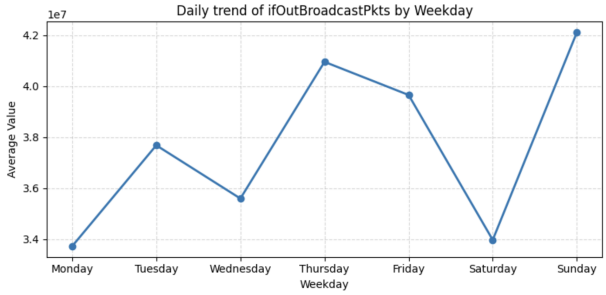


Figure 3: Weekday trend of `ifOutBroadcastPkts`. The x-axis shows weekdays, and the y-axis indicates average outgoing broadcast packet count. The activity pattern oscillates, showing multiple peaks during the week, notably on Thursday and Sunday, with reduced values midweek.

suggests regularly scheduled tasks or periodic data broadcasting events throughout the week, culminating in Sunday peaks.

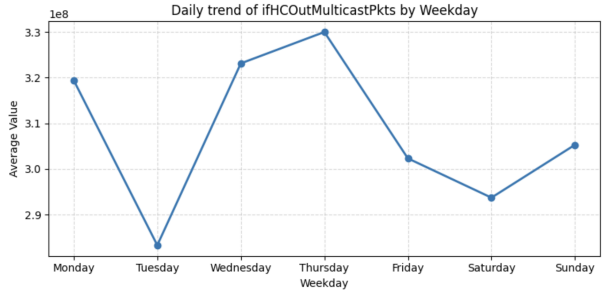


Figure 4: Weekday trend of `ifHCOutMulticastPkts`. The x-axis shows weekdays, and the y-axis represents average outgoing multicast packet count. Activity exhibits alternating peaks and valleys, reaching its highest average on Thursday and lowest on Tuesday.

4. Midweek Variability: Some features, like `ifHCOutMulticastPkts`, demonstrate fluctuating patterns throughout the week with alternating peaks and valleys, notably peaking on Thursday and dipping significantly on Tuesday (Figure 4). This alternating activity may reflect varying load conditions or scheduled multicast traffic tied to specific midweek operational routines.

These weekday-level patterns complement long-term trends and offer additional structure that can inform imputation and forecasting strategies.

3.4 Observed Hourly Trend Patterns

To better understand how network activity fluctuates throughout the day, we examined the hourly average of several telemetry features across the available portion of the dataset. Although our data includes a multi-month gap, aggregating by hour across the full timeline still reveals clear and recurring daily rhythms. Figure 5 highlights three representative patterns observed in different features of the ESNet telemetry data.

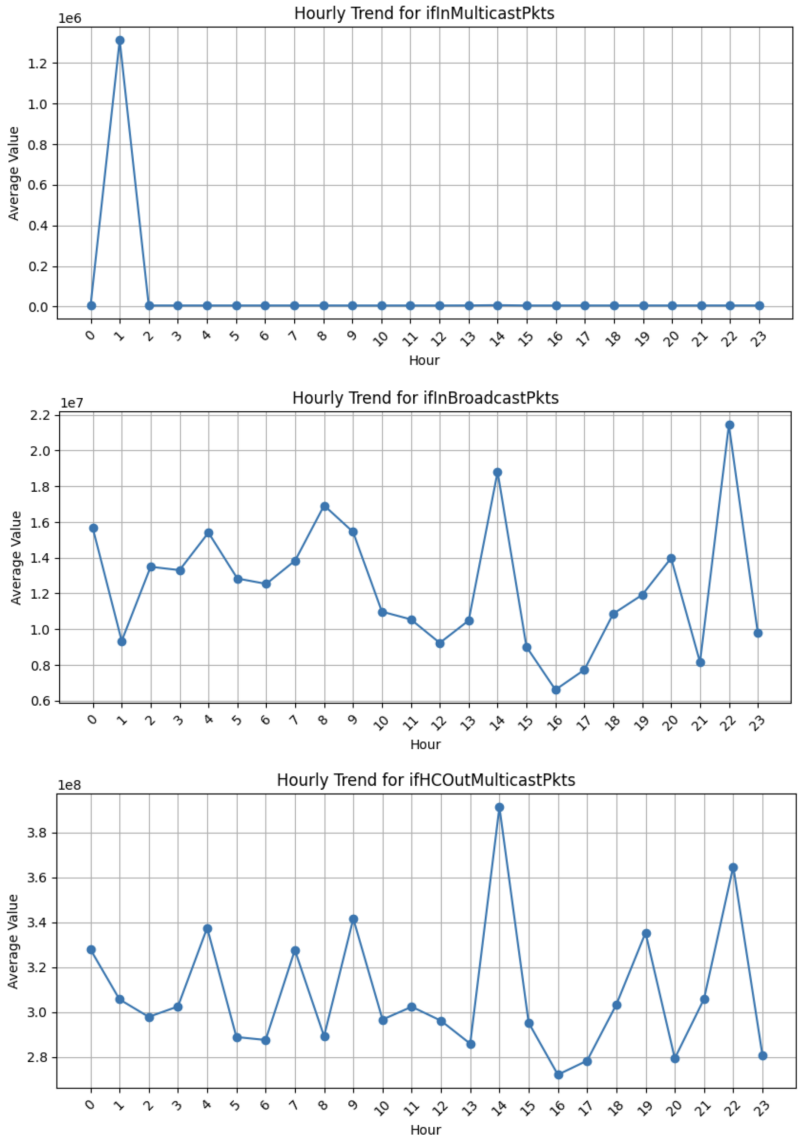


Figure 5: Hourly trends for selected ESNet features. The x-axis shows the hour of the day (0 to 23), and the y-axis shows the corresponding average packet count. Top: `ifInMulticastPkts` shows an extreme spike around 1:00 AM with minimal activity otherwise. Middle: `ifInBroadcastPkts` exhibits multiple fluctuations with notable peaks around midnight, noon, and late evening. Bottom: `ifHCOutMulticastPkts` follows an irregular but repeating pattern with several peaks throughout the day, notably around early afternoon and late evening.

The first pattern, shown in the top panel for `ifInMulticastPkts`, is characterized by minimal activity across most hours, with an isolated and significant spike at around 1:00 AM. This could reflect a scheduled multicast event or a nightly synchronization task.

The second pattern, evident in `ifInBroadcastPkts`, shows variability with multiple peaks at approximately midnight, noon, and especially pronounced peaks late in the evening. This suggests periodic broadcast events or regularly scheduled network operations at these hours.

The third pattern, observed in `ifHCOutMulticastPkts`, demonstrates fluctuating yet repetitive behavior with peaks occurring throughout the day, notably in early afternoon around 14:00 and again around 22:00. Such a rhythm could be related to system maintenance, monitoring routines, or automated multicast transmissions.

Taken together, these examples underscore the diversity of temporal behaviors in the network. Some features reflect human-driven usage spikes, while others hint at automated background processes or asynchronous system operations. These insights are valuable

when designing models that need to learn from or impute over such data, as ignoring time-of-day effects could lead to unrealistic or biased outputs.

4 METHODOLOGY

This study investigates a range of methods for imputing large contiguous gaps in multivariate time series data derived from ESNet telemetry. These gaps, often spanning weeks or months, are common in real-world monitoring systems due to maintenance outages, logging failures, or transient system errors. Standard imputation strategies tend to fail in such contexts, especially when they assume randomly missing values. We systematically compare several established approaches alongside our proposed method, GAPSA (Gap-Aware Probabilistic Simulated Annealing), which is specifically designed to preserve temporal structure and physical constraints in long-gap scenarios.

4.1 Baseline Methods

We benchmark five imputation methods from classical, statistical, and deep learning paradigms:

Forward fill copies the last observed value before the gap forward across all missing time steps. This method is simple but can flatten dynamics and remove temporal variability.

Backward fill mirrors the forward fill approach by propagating the first available value after the gap backward into the missing segment.

Exponential smoothing linearly interpolates between boundary values but applies a decay factor to weigh older observations more heavily, offering a balance between continuity and trend adherence.

BRITS [2] models the time series with a bidirectional recurrent neural network that dynamically estimates missing values as part of the model training loop. Each missing point is treated as a parameter, refined over epochs through backpropagation.

SAITS [5] applies a self-attention mechanism across the temporal and feature dimensions. It relies on a transformer backbone and is capable of learning contextual patterns across long horizons, assuming missingness is sufficiently random and training data is abundant.

4.2 GAPSA: Simulated Annealing for Time Series Imputation

Our method, GAPSA (Gap-Aware Penalized Simulated Annealing), frames the imputation problem as a discrete optimization task over a contiguous missing segment of length N. For each time index i, we construct a small candidate set $\mathcal{X}_i$ of integer values around the expected value $\hat{x}_i$, obtained via interpolation between pre- and post-gap observations. We associate a binary variable $z_{i,x} \in \{0, 1\}$ for each $x \in \mathcal{X}_i$, indicating whether value x is selected at position i.

We minimize the following QUBO objective:

$$
\begin{aligned}
\min_{z} \quad & \sum_{i,x} (x - \hat{x}_i)^2 z_{i,x} \quad \text{(trend fidelity)} \\
& + \sum_{i,x\,:\,x>\max} \gamma_{\text{high}} (x - \max)^4 z_{i,x} \quad \text{(high-value penalty)} \\
& + \sum_{i,x\,:\,x<\min} \gamma_{\text{low}} (\min - x)^2 z_{i,x} \quad \text{(low-value penalty)} \\
& + \sum_{x \in \mathcal{X}_0} (x - y_0 - \delta)^2 z_{0,x} \quad \text{(initial boundary)} \\
& + \sum_{x \in \mathcal{X}_{N-1}} (y_1 - x - \delta)^2 z_{N-1,x} \quad \text{(final boundary)} \\
& + \sum_{i=0}^{N-2} \sum_{\substack{x \in \mathcal{X}_i \\ y \in \mathcal{X}_{i+1}}} ((y - x) - \delta)^2 z_{i,x} z_{i+1,y} \quad \text{(smoothness)} \\
& + \lambda \sum_{i=0}^{N-1} \left(\sum_{x \in \mathcal{X}_i} z_{i,x} - 1 \right)^2 \quad \text{(one-hot constraint)}
\end{aligned}
$$

Here, y_0 and y_1 are the known values immediately before and after the gap, and δ is the estimated per-step increment derived from historical trends. The high- and low-value penalties ensure the imputed values stay within realistic bounds. The final one-hot constraint enforces that exactly one value is chosen per time step.

The resulting QUBO is solved using simulated annealing. If a variable assignment is missing (i.e., all associated $z_{i,x} = 0$), we assign the nearest integer to $\hat{x}_i$ as a fallback. This procedure is repeated independently for each feature in the multivariate series.

4.3 Forecast-based Evaluation

To evaluate the practical impact of imputation, we assess how well the imputed series supports downstream forecasting. We select three forecasting models:

RandomForestRegressor is a non-parametric ensemble model that uses recent lags to predict future values. It offers robustness to outliers but struggles with long-term dependencies.

PatchTST [15] segments the time series into non-overlapping patches and applies self-attention layers to learn temporal representations. It is effective in capturing complex periodicity and scale variations.

DLinear [25] performs time series decomposition followed by simple linear layers. Its inductive bias suits stable or trend-dominated sequences, and it is computationally efficient for long sequences.

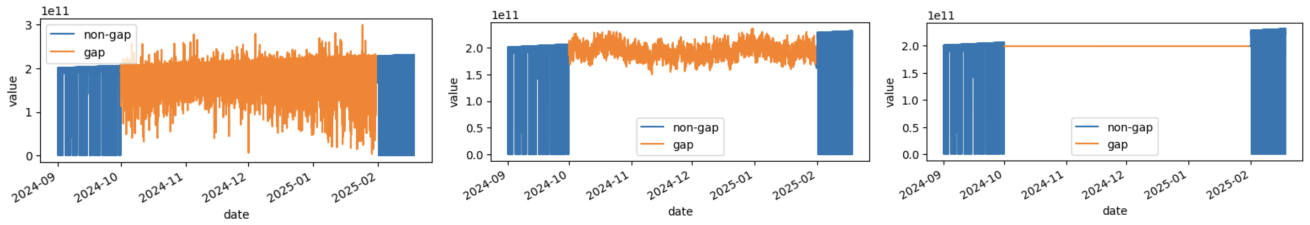
Each forecasting model is trained on data consisting of pre-gap observations and the completed (imputed) gap, then tested on real observed values after the gap. We use MSE (mean squared error) and SMAPE (symmetric mean absolute percentage error) to quantify performance, capturing both magnitude and percentage-based discrepancies.

4.4 Implementation and Reproducibility

All models were implemented in Python using PyTorch and Scikit-learn, with experiments with RandomForestRegressor conducted on an Apple M4 Max CPU. The experiments with DLinear and

Table 1: MSE and SMAPE for various imputation methods across two features. Lower is better.

Feature	Model	Metric	Forward Fill	Backfill	ExpSmooth	SAITS	BRITS	GAPSA
ifHCOutBroadcastPkts	RandomForestRegressor	MSE	2.042	**1.032**	1.144	2.006	1.041	1.675
		SMAPE	191.009	112.115	176.598	196.437	**102.965**	193.035
	PatchTST	MSE	2.286	**1.363**	1.796	1.864	2.545	1.688
		SMAPE	0.676	**0.485**	0.863	1.040	0.623	1.214
	DLinear	MSE	2.285	**1.370**	1.629	1.910	2.598	1.699
		SMAPE	0.698	**0.504**	0.901	1.121	0.699	1.257
ifHCOutOctets	RandomForestRegressor	MSE	1.028	2.187	1.404	1.174	**1.069**	1.399
		SMAPE	167.981	163.701	159.808	173.597	**94.899**	172.757
	PatchTST	MSE	2.285	**1.364**	1.806	1.851	2.545	1.688
		SMAPE	0.676	**0.485**	0.861	1.033	0.623	1.215
	DLinear	MSE	2.285	**1.370**	1.629	1.910	2.598	1.699
		SMAPE	0.698	**0.504**	0.901	1.121	0.699	1.257



(a) GAPSA **(b) SAITS** **(c) BRITS**

Figure 6: Visual comparison of imputation quality for `ifHCOutBroadcastPkts` across three methods. The x-axis shows time (months), and the y-axis indicates outgoing broadcast packet volume. The gap spans October 2024 to January 2025. BRITS yields the lowest MSE and SMAPE (Table 1) but imputes a flat sequence with little variation. SAITS introduces minor fluctuations yet remains overly smooth. In contrast, GAPSA restores realistic variability aligned with pre-gap trends. Although its error is higher numerically, the result is qualitatively more natural, which is especially important for scientific telemetry and bandwidth forecasting tasks.

PatchTST were conducted on an Nvidia Tesla V100 GPU. Our complete codebase is available on GitHub[1] for reproducibility.

5 RESULTS

5.1 ESNet Dataset

Table 1 summarizes the Mean Squared Error (MSE) and Symmetric Mean Average Percentage Error(SMAPE) obtained across different models (RandomForestRegressor, PatchTST, and DLinear) and imputation methods for ifHCOutBroadcastPkts and ifHCOutOctets. The values are reported after standard scaling is applied to the data. In addition, the values for SAITS and BRITS are reported after 100 epochs of training.

Although GAPSA does not achieve the lowest numerical error across all evaluation metrics, qualitative inspection reveals that it produces more plausible and natural imputation sequences. As shown in Figure 6, GAPSA preserves the overall shape and variability of the signal more faithfully than SAITS or BRITS, for example. In contrast, SAITS tends to oversmooth and fill the gap with a single average trend, while BRITS often collapses to a flat extrapolation. These behaviors, while resulting in lower MSE or SMAPE, can obscure critical transitions or anomalies that downstream applications rely on. Our method offers greater interpretability and respects physical constraints such as continuity and counter behavior, which are essential in scientific telemetry contexts.

We also analyze the performance of the forecasting models themselves to understand model efficacy in different data patterns. Across both features, RandomForestRegressor generally performs the worst in terms of SMAPE, particularly on data imputed using SAITS and simulated annealing. This is likely due to the model's limited capacity to capture complex temporal dependencies, which becomes more pronounced when the imputed segments contain high variance or abrupt shifts.

In contrast, PatchTST and DLinear show stronger performance and more interesting variation. For instance, in the ifHCOutBroadcastPkts feature, PatchTST outperforms DLinear on data imputed using backward fill, BRITS, and simulated annealing, while DLinear slightly outperforms PatchTST on exponentially smoothed data. This pattern is consistent with their design philosophies: PatchTST, with its transformer-based patch attention mechanism, handles variability and sudden transitions more robustly, whereas DLinear, relying on decomposition into trend and seasonality, benefits from smoother inputs.

[1] https://anonymous.4open.science/r/KDD_MiLeTs_2025_submission8

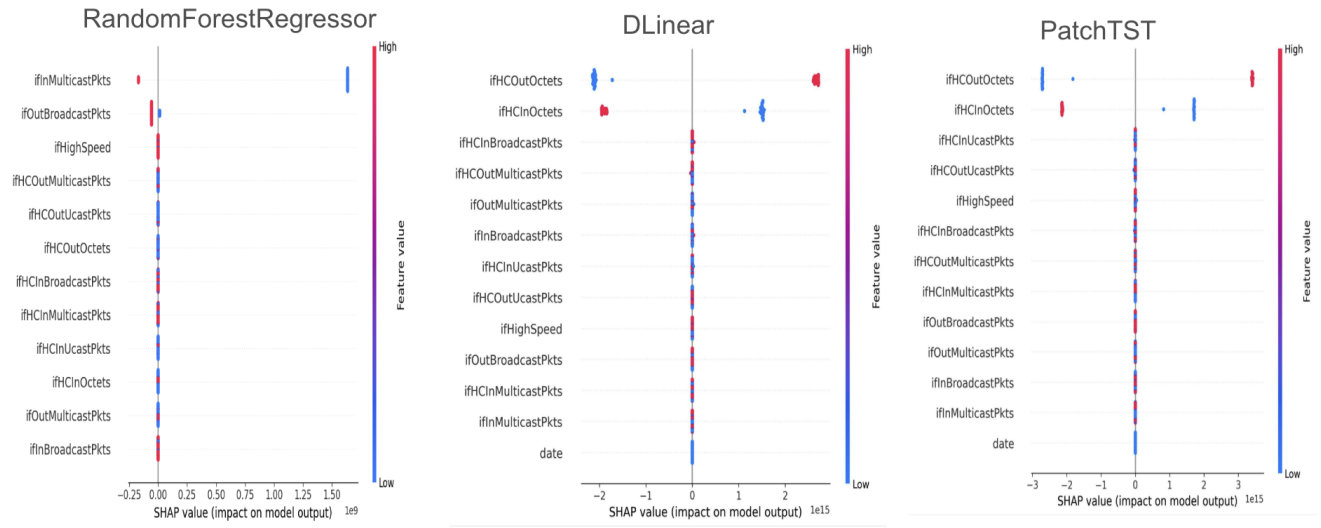


Figure 7: SHAP feature importance plots for `ifHCOutBroadcastPkts` using backward fill imputation. RandomForestRegressor shows minimal and diffuse attributions, while DLinear reveals moderate importance with greater variance. PatchTST exhibits clear separation and strong contributions from key features like `ifHCOutOctets`, reflecting its ability to capture complex temporal dependencies. These differences align with observed forecasting performance, where PatchTST outperforms the others.

The BRITS model performs particularly well across the board, often achieving the lowest MSE and SMAPE among all imputation methods. However, as mentioned earlier, this is not useful as it just produces a flat line over the gap.

Simulated annealing produces more natural-looking imputations, as discussed in the qualitative analysis, but this comes at a slight cost in terms of forecasting accuracy. Its performance remains competitive, particularly with PatchTST, but is often outperformed by BRITS and backward fill in raw metrics. SAITS, while effective on shorter or scattered missing segments, tends to underperform on large contiguous gaps, possibly due to its reliance on learned attention patterns that falter without sufficient temporal context.

Overall, the interaction between imputation method and forecasting model is nontrivial. While backward fill and BRITS produce lower forecasting error on average, methods like simulated annealing offer interpretability advantages, and their compatibility with advanced forecasting models like PatchTST suggests a trade-off worth considering in real-world applications.

In order to analyze the reasoning behind the difference in performance of the forecasting models, we examine the SHAP (SHapley Additive exPlanations) feature importance values for one specific use case: predicting ifHCOutBroadcastPkts with data from the backward fill imputation method.

Figure 7 demonstrates distinct differences in feature utilization and model interpretability across the three models. Notably, the ifHCOutBroadcastPkts feature exhibits varying levels of impact.

RandomForestRegressor displays relatively small SHAP magnitudes across all features, with values tightly clustered around zero. This indicates that the model fails to extract strong and consistent patterns from the input features, consistent with the known limitations of tree-based methods in high-dimensional or highly correlated time series settings.

In contrast, DLinear and PatchTST show substantially larger SHAP magnitudes. DLinear captures stronger feature relationships compared to RandomForestRegressor, but its SHAP values exhibit greater variance, suggesting sensitivity to input noise. PatchTST, however, exhibits the sharpest feature differentiation, with high SHAP magnitudes and well-separated impacts across features such as ifHCOutOctets and ifHCInOctets. This sharper feature separation implies that PatchTST more effectively captures both short-term fluctuations and long-term dependencies.

Overall, these observations align with the model performance outcomes: PatchTST outperforms both DLinear and RandomForestRegressor, with DLinear occupying an intermediate position. The superior performance of PatchTST can be attributed to its capacity to model complex temporal feature interactions, whereas RandomForestRegressor's underperformance highlights the challenges of applying traditional tree-based methods to irregular and high-dimensional time series data.

5.2 Air Quality Dataset

To further evaluate the generalizability of our imputation approach, we applied it to the widely-used Air Quality dataset. A synthetic 7-day gap was inserted in the multivariate time series, and each imputation method was tasked with filling this gap. All experiments were conducted on an Apple M4 Max CPU with 64 GB of memory. For fairness, SAITS and BRITS were trained for 150 epochs each.

Table 2 summarizes the performance across four key metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the wall-clock time

required to perform the imputation. It should be noted that these values are reported without standard scaling. Our simulated annealing approach achieved the lowest RMSE (294.6), MAE (156.3), and MAPE (85.1), while completing in just under 21 seconds. In contrast, BRITS and SAITS had RMSEs of 645.5 and 628.0, and MAPE values exceeding 100%, with runtimes of over 9 and 8 minutes respectively.

Although BRITS and SAITS were given ample time to train, their results were still less accurate than those from our method. It's worth noting, however, that these models might improve with longer training or fine-tuned hyperparameters. Still, the results suggest that our optimization-based approach can produce strong and efficient imputations without relying on large-scale training or GPU acceleration.

Table 2: Imputation Performance on the Air Quality Dataset

Model	RMSE	MAE	MAPE	Time (s)
GAPSA	294.6	156.3	85.1	**20.95**
BRITS (150 epochs)	645.5	423.1	97.4	1407.5
SAITS (150 epochs)	628.0	405.7	106.1	538.5

6 CONCLUSION AND FUTURE WORK

This study introduced a physics-inspired imputation framework using simulated annealing to address large, contiguous gaps in time series data. By formulating the imputation task as a constrained optimization problem, our approach explicitly integrates domain knowledge—such as periodic trends and monotonic counters—into a unified QUBO model. Through extensive evaluation on network telemetry and the Air Quality dataset, we demonstrated that GAPSA yields competitive or superior imputation quality compared to state-of-the-art learning-based models, particularly in terms of preserving inherent variability of the time series with limited computing resources.

While deep learning models like BRITS and SAITS can outperform in certain metrics given sufficient training time and tuning, they require significantly higher runtimes and depend heavily on GPU acceleration. In contrast, our approach achieves reliable performance with minimal configuration and can run efficiently on standard CPUs.

Looking ahead, one promising direction is to accelerate our optimization-based approach using quantum annealing [8] hardware. Since our formulation naturally maps to the QUBO format, it is well-suited for quantum processors like D-Wave's Advantage system. Leveraging quantum annealing could allow us to solve larger imputation problems in less time or with better solution quality, especially for high-dimensional datasets. Future work will also explore hybrid approaches that combine physics-based regularization with deep learning models to improve generalization while maintaining interpretability.

Ultimately, this work highlights the value of tailored imputation strategies in critical domains like scientific networking, where data gaps are long, structure matters, and trustworthiness is paramount.

ACKNOWLEDGMENTS

This work is supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357

REFERENCES

[1] B. Abi et al. 2020. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I: Introduction to DUNE. arXiv:2002.02967 [physics.ins-det] https://arxiv.org/abs/2002.02967

[2] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. arXiv:1805.10572 [cs.LG] https://arxiv.org/abs/1805.10572

[3] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports* 8, 1 (2018), 6085.

[4] DOE. 2023. The Tachyon Project: Scalable Modeling Framework for HEP. https://tachyon-org.github.io/

[5] Wenjie Du, David Côté, and Yan Liu. 2023. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications* 219 (June 2023), 119619. https://doi.org/10.1016/j.eswa.2023.119619

[6] Zoubin Ghahramani and Geoffrey E Hinton. 1996. Parameter estimation for linear dynamical systems. *University of Toronto Technical Report CRG-TR-96-2* (1996).

[7] Rob J. Hyndman and George Athanasopoulos. 2008. *Forecasting: Principles and Practice*. OTexts, Sydney, Australia. https://otexts.com/fpp3/

[8] Tadashi Kadowaki and Hidetoshi Nishimori. 1998. Quantum annealing in the transverse Ising model. *Physical Review E* 58, 5 (Nov. 1998), 5355–5363. https://doi.org/10.1103/physreve.58.5355

[9] Ina Khandelwal, Ratnadip Adhikari, and Ghanshyam Verma. 2015. Time Series Forecasting Using Hybrid ARIMA and ANN Models Based on DWT Decomposition. *Procedia Computer Science* 48 (2015), 173–179. https://doi.org/10.1016/j.procs.2015.04.167 International Conference on Computer, Communication and Convergence (ICCC 2015).

[10] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200209.

[11] Roderick JA Little and Donald B Rubin. 2019. Statistical analysis with missing data. *Wiley Series in Probability and Statistics* (2019).

[12] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30 (2017), 4765–4774.

[13] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research* 11, 80 (2010), 2287–2322. http://jmlr.org/papers/v11/mazumder10a.html

[14] Fulufhelo V. Nelwamondo, Shakir Mohamed, and Tshilidzi Marwala. 2007. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science* 93, 11 (2007), 1514–1521. http://www.jstor.org/stable/24099079

[15] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. arXiv:2211.14730 [cs.LG] https://arxiv.org/abs/2211.14730

[16] OpenAI. 2024. ChatGPT (May 28 version). https://chat.openai.com. Accessed via https://chat.openai.com.

[17] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. 2019. Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/3a0844cee4fcf57de0c71e9ad3035478-Paper.pdf

[18] Waleed Softah, Laleh Tafakori, and Hui Song. 2025. Analyzing and predicting residential electricity consumption using smart meter data: A copula-based approach. *Energy and Buildings* 332 (2025), 115432. https://doi.org/10.1016/j.enbuild.2025.115432

[19] Daniel J. Stekhoven and Peter Bühlmann. 2011. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics* 28, 1 (10 2011), 112–118. https://doi.org/10.1093/bioinformatics/btr597 arXiv:https://academic.oup.com/bioinformatics/article-pdf/28/1/112/50568519/bioinformatics_28_1_112.pdf

[20] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. arXiv:2107.03502 [cs.LG] https://arxiv.org/abs/2107.03502

[21] Stef van Buuren and Karin Groothuis-Oudshoorn. 2011. MICE: Multivariate imputation by chained equations in R. In *Journal of Statistical Software*, Vol. 45. 1–67.

[22] Saverio Vito. 2016. Air Quality. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C59K5F.

[23] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2018. GAIN: Missing Data Imputation using Generative Adversarial Nets. arXiv:1806.02920 [cs.LG] https://arxiv.org/abs/1806.02920

[24] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal Regularized Matrix Factorization for High-dimensional Time Series Prediction. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2016/file/85422afb467e9456013a2a51d4dff702-Paper.pdf

[25] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are Transformers Effective for Time Series Forecasting? arXiv:2205.13504 [cs.AI] https://arxiv.org/abs/2205.13504