Fusion-MTSI: Fusion-Based Multivariate Time Series Imputation

Sangyong Lee [1,*] and Subo Hwang [2]

[1] AI Research Center, OKESTRO Co., Ltd., Seoul, Republic of Korea
[2] Department of Physics and Astronomy, Seoul National University, Seoul, Republic of Korea
Email: sangyong1996@gmail.com (S.Y.L.); sbhwang@snu.ac.kr (S.B.H.)
*Corresponding author

***Abstract*—Missing data poses a significant challenge in multivariate time series, disrupting continuity and leading to biases in analysis. Addressing these gaps is essential, as incomplete data can undermine the reliability of models in various applications. To overcome this challenge, we propose Fusion-Based Multivariate Time Series Imputation (Fusion-MTSI), a novel imputation method that addresses the limitations of traditional approaches, which often struggle to capture complex temporal and cross-feature relationships. Fusion-MTSI overcomes this limitation by leveraging both feature-wise and point-wise comparisons, enabling the detection of broad patterns and subtle temporal variations across features. By relying only on the intrinsic characteristics of data, Fusion-MTSI achieves effective imputation without requiring domain-specific knowledge. Experimental results across six real-world datasets demonstrate that Fusion-MTSI outperforms conventional methods, achieving up to a 31% reduction in Mean Squared Error, making it a robust, adaptable choice for diverse applications.**

***Keywords*—missing data imputation, multivariate time series, feature relation, temporal causality, machine learning**

I. Introduction

Time series data encompasses information observed or measured over time and serves as a critical decision-making tool across industries and academic disciplines [1]. In industrial applications, time series data is widely used to enhance productivity and reduce costs in domains such as retail [2], finance [3], manufacturing [4], telecommunications [5], and big data analytics [6]. Academically, time series data facilitates solving problems such as economic forecasting, understanding natural phenomena, disaster prevention, and early disease diagnosis through temporal pattern analysis in fields like economics [7], astronomy [8], geology [9], and environmental science [10]. Moreover, in areas like climate modeling [11], biology [12], and medicine [13], time series data plays a key role in quantitatively analyzing complex and dynamic phenomena and predicting future trends. The widespread adoption of sensors and IoT devices has led to an exponential growth in the quantity and diversity of measurable time series data. This surge highlights the pressing demand for systems and analytical methods to effectively analyze and utilize time series data.

In this context, time series data collected in real-world industrial settings is often in the form of Multivariate Time Series (MTS), which consists of multiple interacting variables. While MTS structure provides rich information and enhances the accuracy of analyses, it also introduces significant complexity in data processing and modeling. Notably, missing data due to sensor malfunctions or maintenance can reduce the accuracy of analysis and compromise the consistency and reliability of the data [14, 15]. Consequently, techniques capable of effectively inputting missing values are indispensable for maximizing the value of MTS data. It is particularly crucial to develop methods that preserve the multidimensional characteristics of the data while addressing missingness.

To address this issue, a wide range of approaches for handling missing values has been studied. Traditional statistical and deterministic methods [1] include simple imputation techniques (e.g., zero, mean, and last observation carried forward) as well as time series models such as Autoregressive Integrated Moving Average (ARIMA), Autoregressive Fractionally Integrated Moving Average (ARFIMA), and seasonal ARIMA (SARIMA). Interpolation methods like linear interpolation and cubic spline interpolation are also commonly used [16]. Iterative model-based methods such as MissForest [17] and Multivariate Imputation by Chained Equations (MICE) [18] have gained attention. Additionally, KNN-based methods such as *k*-Nearest Neighbor Imputation (kNNI) [19], Dynamic Time Warping *k*-Nearest Neighbor (DTW-kNN) [20], and Fuzzy K-Nearest Neighbor (FKNN) [21] have been proposed as approaches.

KNN-based methods estimate missing values by identifying the *k*-nearest neighbors of observation with missing values and utilizing their information for imputation. To reflect the temporal nature of time series data, existing approaches often employ Dynamic Time Warping (DTW) [22] instead of Euclidean distance or incorporate temporal lags with non-linear relationships using Spearman correlation. DTW-kNN uses DTW for distance measurement and Pearson correlation coefficients

Manuscript received November 22, 2024; revised January 2, 2025; accepted February 26, 2025; published May 9, 2025.

as feature weights to interpolate missing segments. FKNN utilizes Euclidean distance rather DTW and leverages Spearman correlation to capture non-linear relationships among variables while focusing on subsets of features with temporal lags. However, these methods have notable trade-offs. DTW-kNN's reliance on Pearson correlation limits its ability to capture complex global non-linear relationships [23]. Additionally, the performance of DTW-kNN is highly dependent on window size, demanding dataset-specific tuning, which hinders generalizability. In contrast, FKNN leverages Spearman correlation to capture global non-linearities. However, its reliance on Euclidean distance makes it less effective in handling local patterns with consecutive missing data. Addressing local pattern limitations often requires domain-specific expertise to fine-tune hyperparameters. Thus, an ideal imputation approach for MTS data should effectively balance global non-linear relationships across variables and local patterns when handling consecutive missing data.

In this paper, we propose Fusion-Based Multivariate Time Series Imputation (Fusion-MTSI), a novel methodology that resolves the trade-offs between existing methods and overcomes their shared challenges. Fusion-MTSI captures global relationships by leveraging Spearman correlation to model non-linear inter-variable dependencies. In addition, Fusion-MTSI models local relationships by utilizing past and future time points of variables to effectively handle consecutive missing segments. Furthermore, the proposed method reduces sensitivity to window size, ensuring robust performance across diverse datasets. By integrating global and local capabilities, this approach addresses the limitations of existing methods, offering a comprehensive approach to modeling the intricate nature of time series data. The main contributions of this paper are the following:

- Fusion-MTSI captures both global and local trends in MTS data through feature-wise and point-wise comparisons. This dual capability enables more accurate imputation by extracting relevant features and accounting for temporal causality.
- Fusion-MTSI effectively utilizes the intrinsic structure of the data, reducing dependence on domain knowledge. This makes it a more generalizable approach to MTS data across various domains, especially useful in cases where domain expertise is limited.
- In experimental results, Fusion-MTSI shows enhanced performance and robustness for missing value imputation across six real-world MTS datasets, outperforming existing methods.

The paper is organized as follows. Section II explores the mechanisms of existing methods that form the foundation of our proposed approach. Section III provides a detailed explanation of the proposed method. Section IV presents both quantitative and qualitative evaluations. Finally, Section V summarizes our contributions and outlines directions for future research.

II. Related Work

A. Imputation Methods

1) Deterministic and statistical methods for imputation

Imputation methods for time series data include deterministic approaches such as linear interpolation and cubic spline interpolation [1], as well as statistical techniques like Moving Average (MA), which predicts missing values based on the mean within a sliding window, and extended methods like ARIMA, ARFIMA, and SARIMA. Deterministic methods use explicit rules to fit missing segments with linear or cubic functions, resulting in low computational complexity. However, their simplicity often results in suboptimal accuracy, particularly for complex patterns. In contrast, statistical methods leverage the statistical properties of the data, offering improved accuracy over deterministic approaches. Statistical methods, however, are inherently designed for univariate time series, which restricts their use with multivariate data.

2) Iterative model-based imputation

Iterative model-based imputation methods have been developed to improve the accuracy and reliability of imputed values. For instance, MissForest [17] uses a random forest model iteratively trained to impute missing values. Similarly, MICE [18] employs linear regression models iteratively, using data from other features to provide optimal estimates. However, the performance of iterative models heavily depends on the nature of the missing data (e.g., Missing at Random (MAR), Missing Completely at Random (MCAR), Missing Not at Random (MNAR)) [24, 25] and the specific random forest or regression model employed. Additionally, these methods do not inherently account for the similarity between features. As a result, even features with highly divergent trends are given equal weighting during imputation, regardless of their relevance.

3) KNN-based imputation

Several KNN-based imputation methods have been proposed to address missing data challenges [19, 20, 21, 26]. The most basic method, KNNI [19], uses Euclidean distance to measure the similarity between time series of different features. It then selects the k nearest features and imputes missing values using their mean value. When using Euclidean distance as the metric, comparisons are restricted to matching time points, requiring time series of equal length. To overcome this constraint, alternative approaches have been proposed that replace Euclidean distance with Dynamic Time Warping (DTW) [20, 26].

For instance, DTW-kNN [20] eliminates the requirement for equal-length time series and allows comparisons across time-shifted data points through temporal alignment. DTW-kNN further assigns feature importance weights using Pearson correlation coefficients, assigning higher priority to values from highly similar features during imputation. However, DTW-kNN struggles to capture complex global nonlinear relationships due to its reliance on Pearson correlation and

is highly sensitive to window size and requires dataset-specific tuning.

On the other hand, FKNN [21] does not utilize DTW but replaces Pearson correlation with Spearman correlation to better model nonlinear inter-variable relationships, prioritizing highly relevant features. FKNN uses Euclidean distance and leverages domain knowledge to address temporal lags, leveraging sliding window information related to missing points. However, FKNN faces difficulties in handling consecutive missing patterns due to its dependence on Euclidean distance. Additionally, FKNN relies on domain expertise for determining optimal window sizes and dictionary sets. Consequently, both methods face challenges with generalization across diverse datasets.

B. Distance between Different Time Series

1) Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) [22] is a method for measuring the distance between two time series, offering the flexibility to compare time series with time shifts or different lengths. The primary objective of DTW is to determine the optimal alignment between two time series by performing point-wise matching of their data points. This is achieved by stretching or compressing the time axis to align each component of one time series with the other. To calculate the DTW distance, a distance measure between two points must be defined. Since this choice is arbitrary, in this paper, we adopt the absolute difference between the values as the distance metric.

2) Move-Split-Merge (MSM) distance

The Move-Split-Merge (MSM) distance [27] is a metric that quantifies the distance between two time series by calculating the minimum cost required to transform one time series into another. Like DTW, MSM can quantitatively measure the dissimilarity between time series. The MSM satisfies all three properties of a mathematical metric: reflexivity, symmetry, and the triangle inequality, while DTW does not satisfy the triangle inequality. The MSM distance is computed using three transformation operations: Move, Split, and Merge. The Move operation is associated with a cost defined by the difference in values before and after the move. The costs for the Split and Merge operations are determined by a user-defined parameter c. Among all possible transformation paths, the MSM distance is defined as the minimum cost path, measuring the degree of transformation between the two time series.

III. Methods

In this study, we propose a novel missing value imputation method called Fusion-MTSI, which simultaneously considers the global relationships among different features in multivariate time series and the local similarities across distinct time steps. To capture the global relationships, we employ Spearman correlation coefficient to quantify the overall trends among variables, and utilize feature-wise DTW [22] and MSM [27] to measure the distance between two time series. Meanwhile, local relationships are incorporated by performing point-wise comparisons each feature to reflect similarities across different time points. In this paper, we represent a multivariate time series dataset with m features observed over the time interval $\{1, 2, \dots, n\}$. Let the observation vector at time t be denoted by $f_t = (f_t^1, f_t^2, \dots, f_t^m)^\top$, which can be arranged into an $m \times n$ matrix S. In practice, some entries in this matrix may be missing. We refer to any feature that contains missing values as a "target feature".

Fusion-MTSI is composed of three main steps. In the first step, Feature Extraction, we identify variables that exhibit similar patterns to the target features. In the second step, Relationship Weight Calculation, we incorporate both global relationships (via feature-wise similarity) and local relationships (via point-wise similarity) to compute the relationship weights among variables and time points. Finally, in the Imputation step, we use these computed weights to fill in the missing values. By effectively combining correlations among different variables (global relationships) with similarities at specific time points (local relationships), Fusion-MTSI achieves more accurate and reliable missing value imputation in multivariate time series.

A. Feature Extraction

Feature extraction is the process of extracting variables that exhibit globally similar patterns to a target feature. First, we designate the i-th variable—one that contains missing values—as the target feature f^i. We then extract features whose global patterns closely resemble f^i. To facilitate this, we perform a temporary linear interpolation of missing segments only during the feature extraction stage. Next, we compute the Spearman correlation coefficient s_{ij} between the target feature f^i and the j-th variable. We select the top k_f variables whose absolute correlation values $|s_{ij}|$ are the highest. If any selected variable has a negative s_{ij}, we apply the transformation $1-f_t^j$ to all time indices $t \in \{1, 2, \dots, n\}$ to maintain a positive correlation direction. Variables identified in this manner for each target feature are referred to as reference features. Fig. 1 provides a visual overview of this reference feature extraction process.

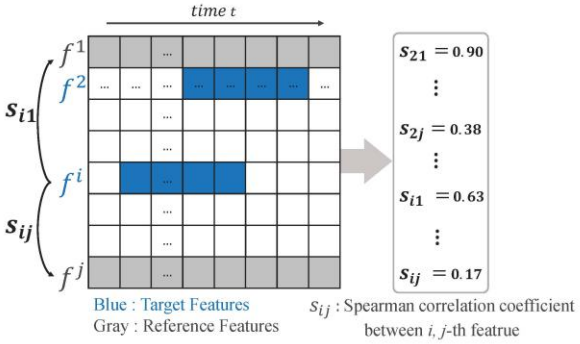

Fig. 1. Feature extraction procedure of i-th target feature related with j-th reference feature.

B. Relationship Weight Calculation

Once the top-k_f reference features most similar to each target feature have been identified, we proceed to compute the relationship weights across variables and time series.

1) Global relationship

The global relationship weight G_{ij} is derived solely from the target feature and its reference features. G_{ij} reflects the similarity between two variables such that more similar variables receive higher weights. To achieve this, we first calculate D_{ij}, which quantifies the global distance between two time series.

D_{ij} denotes the global distance between the target feature f^i and a reference feature f^j. Even if the two time series differ in length due to missing values, D_{ij} is computed as follows:

$$D_{ij} = \log\{\mathrm{DTW}(f^i, f^j) + 1\} + \log\{\mathrm{MSM}(f^i, f^j) + 1\} \quad (1)$$

$\mathrm{DTW}(f^i, f^j)$ is the Dynamic Time Warping distance and $\mathrm{MSM}(f^i, f^j)$ is the Move-Split-Merge distance. A lower D_{ij} value implies a higher degree of similarity between the two time series.

Using the calculated D_{ij}, we determine the global relationship weight G_{ij} as follows:

$$G_{ij} = \frac{|s_{ij}|^{\alpha}}{D_{ij}} \quad (2)$$

s_{ij} is the Spearman correlation coefficient, and α is the Spearman exponent, a parameter that adjusts the influence of the correlation coefficient. This formulation combines both correlation and global distance information, resulting in a larger G_{ij} for more similar pairs of variables.

2) Local relationship

The local relationship weight captures the temporal variability of reference features. To compute The local relationship weight, we first identify the missing time point l in the target feature and a reference time point p, then measure the variability between these two time points (l, p) within each reference feature.

By directly comparing the variability at the time point requiring imputation with the observed data, the local relationship weight facilitates more precise missing value recovery. Let L_{lp} be computed as follows:

$$L_{lp} = \left[\frac{\sqrt{\sum_j^{k_f}\left(f_l^j - f_p^j\right)^2}}{\sum_p^n \sqrt{\sum_j^{k_f}\left(f_l^j - f_p^j\right)^2}}\right]^{-1} \quad (3)$$

Conceptually, L_{lp} represents the variability between two time points in one time series. When the variability is small, L_{lp} becomes large, thereby assigning a higher weight.

Eqs. (2) and (3) illustrate the operations in a row-to-row perspective and a column-to-column perspective, respectively, when treating the target and reference features as matrices. Fig. 2 provides a visual representation of these operations. In the row-to-row computation, the target feature is compared against each reference feature across the entire time span to obtain global relationship weights that capture overall characteristics. In contrast, the column-to-column computation compares all available reference feature values at a specific time with the missing time point, thereby reflecting local characteristics and enabling the calculation of local relationship weights.

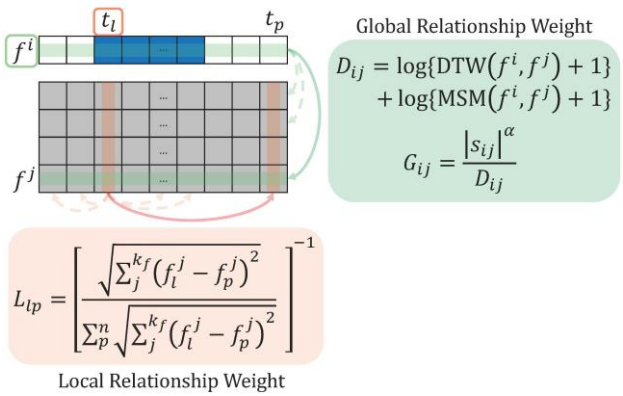

Fig. 2. Fusion-MTSI Relationship weight calculation procedure.

Finally, by taking into account both the global relationship weight G_{ij} and the local relationship weight L_{lp}, we retrieve the missing time point index f_l from the target feature and compute its difference with the reference time point index f_p in the reference features. Based on these differences, we define a vector called the Fusion difference $V \in \mathbb{R}^n$, whose p-th component is expressed as follows:

$$V_p = \sqrt{\sum_j^{k_f} \frac{\left(f_l^j - f_p^j\right)^2}{L_{lp} G_{ij}}} \quad (4)$$

V_p is a difference value computed by integrating both the global and local relationship weights; it decreases as the two weights increase. In other words, stronger relationships in either weight lead to a relatively smaller distance between the two time points.

C. Imputation

Leveraging a KNN Regressor at its core, Fusion-MTSI selects the k elements with the smallest Fusion difference values. Specifically, it computes the weight w_p from the reciprocal of the Fusion difference and assigns greater weight to indices with larger w_p. The final imputation estimate produced by Fusion-MTSI can be expressed as follows:

$$f_l^i = \sum_p w_p f_p^i \quad (5)$$

Algorithm 1 provides a concise summary of the entire computational procedure for Fusion-MTSI.

Algorithm 1. Fusion-MTSI Imputation

Require: Scaled matrix $S \in \mathbb{R}^{m \times n}$, number of reference features k_f, Spearman exponent α, number of nearest neighbors k.

Ensure: Imputed matrix $\bar{S}_{\text{filled}}$

Initialize $\bar{S} \leftarrow S$ with missing values masked;
for each target feature f^i in $\bar{S}$ **do**
  Linearly impute missing values in f^i;
end for
for each target feature f^i in $\bar{S}$ **do**
  for each feature f^j in $\bar{S}, j \neq i$ **do**
    Compute Spearman correlation coefficient s_{ij} of (f^i, f^j);
    $s_{ij} \leftarrow |s_{ij}|$;
  end for
  Select top-k_f features as reference features;
  for each reference feature f^j **do**
    if original s_{ij} is negative **then**
      Transform $f^i \leftarrow 1 - f^j$;
    end if
  end for
  Compute similarity D_{ij} for each reference feature f^j;
    $D_{ij} \leftarrow \log\{\text{DTW}(f^i, f^j) + 1\} + \log\{\text{MSM}(f^i, f^j) + 1\}$;
  Compute global relationship weight G_{ij} for each f^j;
    $G_{ij} \leftarrow \frac{|s_{IJ}|^\alpha}{D_{ij}}$;
  for each missing value at time index l in f^i **do**
    for each time index p where f_p^i is observed **do**
      Compute local relationship weight L_{lp};
      $L_{lp} \leftarrow \left[\frac{\sqrt{\sum_j^{k_f}(f_l^j - f_p^j)^2}}{\sum_p^n \sqrt{\sum_j^{k_f}(f_l^j - f_p^j)^2}}\right]^{-1}$;
      Compute Fusion difference V_p;
      $V_p \leftarrow \sqrt{\sum_j^{k_f} \frac{(f_l^j - f_p^j)^2}{L_{lp} G_{ij}}}$;
    end for
    Select k indices of smallest V_p;
    Compute weights $w_p \leftarrow \frac{1}{V_p}$ for selected indices;
    Normalize weights: $w_p \leftarrow \frac{w_p}{\sum w_p}$;
    Impute $f_l^i \leftarrow \sum w_p f_p^i$;
  end for
end for
return imputed matrix $\bar{S}_{\text{filled}}$

IV. Experimental Results

A. Datasets

To evaluate the effectiveness of the proposed method, Fusion-MTSI, we utilized six real-world datasets. Detailed descriptions of the datasets are as follows:

- **Electricity**[1]: Consists of the hourly electricity consumption data of 321 customers over the period from 2012 to 2014.
- **Exchange** [29]: This dataset records the daily exchange rates of eight different countries, covering the period from 1990 to 2016.
- **Traffic**[2]: A collection of hourly road occupancy rates from the California Department of Transportation, measured by sensors on freeways in the San Francisco Bay area.
- **Weather**[3]: Includes 21 meteorological indicators such as air temperature and humidity, recorded every 10 minutes for the entire year of 2020.
- **ILI**[4]: Contains weekly data of influenza-like illness (ILI) cases from the Centers for Disease Control and Prevention (CDC) in the United States, spanning from 2002 to 2021.
- **ETT***: Contains data collected from electricity transformers, specifically load and oil temperature. It includes hourly recorded data (ETTh1 and ETTh2) and 15-minute interval data (ETTm1 and ETTm2) from July 2016 to July 2018. In our results, the values are the average across all four versions: ETTm1, ETTm2, ETTh1, and ETTh2.

B. Implementation Details

During our experiments, we compare five baseline methods for imputation, including the basic interpolation technique Linear, the nearest-neighbor based method Nearest, KNNI [20], DTW-Knn [21], and FKNN [22]. For these experiments, we set the default parameters as follows: the missing rate $r_{missing} = 0.05$, representing the ratio of the total number of data points to the number of missing values, and the maximum length of consecutive missing values $l_{missing} = 0.05$.

In addition, we report results for the best-performing nearest-neighbor-based imputation technique using various values for the number of neighbors: $k_n \in \{1, 3, 5, 10, 50, 100\}$. For our proposed method, we also present the best-performing results based on the number of most relevant features, with $k_f \in \{3, 5, 7, 10\}$.

C. Quantitative Results

Table I summarizes the results across six datasets. Fusion-MTSI demonstrates significant Mean Squared Error (MSE) reductions across five datasets, achieving up to a 59% improvement on Weather and an average

1 https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

2 http://pems.dot.ca.gov/

3 https://www.bgc-jena.mpg.de/wetter/

4 https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

reduction of 31% across all datasets (Electricity, Exchange, Traffic, Weather, ILI, ETT*). On the Exchange dataset, characterized by a lack of clear periodicity, Fusion-MTSI achieves a 20% improvement (from 0.0031 to 0.0025) over other baselines. Interestingly, the Linear method performs best on the Exchange dataset, likely due to its effective handling of non-stationary economic data.

TABLE I. EXPERIMENTAL RESULTS ON SIX REAL-WORLD BENCHMARK DATASETS

Dataset Metric	Electricity		Exchange		Traffic		Weather		ILI		ETT*	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Linear	0.0193	0.1021	**0.0024**	**0.0338**	0.0140	0.0823	0.0118	0.0805	0.0319	0.1228	0.0083	0.0618
Nearest	0.0233	0.1096	0.0031	0.0392	0.0407	0.1237	0.0144	0.0892	0.0168	0.1237	0.0110	0.0702
MICE	0.0030	0.0414	0.0064	0.0678	0.0036	0.0509	0.0034	0.0504	0.0047	0.0585	0.0020	0.0298
MissForest	0.0039	0.0429	0.0052	0.0525	0.0017	0.0248	0.0025	0.0349	0.0065	0.0605	0.0026	0.0372
KNNI	0.0029	0.0351	0.0038	0.0476	0.0015	0.0171	0.0017	0.0250	0.0035	0.0272	0.0020	0.0266
DTW-kNN	0.0126	0.0888	0.0034	0.0463	0.0081	0.0633	0.0087	0.0702	0.0113	0.0677	0.0064	0.0579
FKNN	0.0029	0.0331	0.0035	0.0469	0.0012	0.0154	0.0016	0.0218	0.0034	0.0272	0.0017	0.0248
Fusion-MTSI	**0.0026**	**0.0324**	0.0025	0.0371	**0.0011**	**0.0145**	**0.0007**	**0.0101**	**0.0017**	**0.0204**	**0.0012**	**0.0192**

However, Linear struggles to capture the complex temporal patterns found in real-world series. While the performance gap between Linear and Fusion-MTSI on the Exchange dataset is minimal (~2%), Fusion-MTSI demonstrates consistently strong and reliable performance across other datasets. This highlights Fusion-MTSI's robust imputation capabilities in both stationary and non-stationary environments, making it highly suitable for real-world applications.

In Table II, we further evaluate the imputation accuracy on the Weather dataset, which is characterized by intricate inter-variable dependencies, dynamic spatiotemporal changes, and highly non-linear patterns. This evaluation focuses on how well the imputed data preserves the underlying correlations between variables compared to the original dataset. The imputation performance is quantitatively assessed using standard metrics such as MSE and Mean Absolute Error (MAE). Additionally, the correlation preservation is measured by computing the Mean Percent Difference (%) and Max Percent Difference (%) between the original and imputed data, providing insights into the overall quality of imputation.

TABLE II. DIFFERENCES IN CORRELATION BETWEEN VARIABLES OF RESTORED DATA AND ORIGINAL DATA

Method	MSE	MAE	Mean Percent Difference (%)	Max Percent Difference (%)
DTW-kNN	0.0087	0.0702	10.1417	489.7826
Fusion-MTSI (Pearson)	**0.0029**	**0.0315**	**5.2459**	**261.7590**
FKNN	0.0016	0.0218	3.0050	147.2458
Fusion-MTSI (Spearman)	**0.0007**	**0.0101**	**0.8068**	**21.5574**

We compared the imputation performance of DTW-kNN, Fusion-MTSI (Pearson), FKNN, and Fusion-MTSI (Spearman). Both DTW-kNN and Fusion-MTSI (Pearson) utilize the Pearson correlation coefficient to analyze inter-variable relationships. However, their limitations in capturing complex non-linear relationships were evident when evaluated using Mean Percent Difference and Max Percent Difference. Notably, Fusion-MTSI (Pearson) consistently outperformed DTW-kNN, demonstrating greater robustness and effectiveness in reflecting local patterns. While DTW-kNN is highly sensitive to window size, requiring dataset-specific tuning, Fusion-MTSI (Pearson) mitigates this limitation. Across all metrics, including MSE, MAE, Mean Percent Difference, and Max Percent Difference, Fusion-MTSI (Pearson) achieved superior performance.

Methods capable of capturing non-linearity, such as FKNN and Fusion-MTSI (Spearman), demonstrated superior performance compared to Pearson-based methods, achieving lower Mean Percent Difference and Max Percent Difference values. Notably, Fusion-MTSI (Spearman) outperformed FKNN by effectively capturing local relationships without the limitations of window size. This advantage arises from Fusion-MTSI's ability to simultaneously model global non-linear interactions among variables and local temporal dynamics, enabling more robust imputation across diverse scenarios.

In conclusion, Fusion-MTSI effectively overcomes the limitations of existing methods, such as sensitivity to window size and insufficient reflection of non-linearity. By integrating both global and local relationships, Fusion-MTSI enables more precise and accurate imputation of multivariate time series data, overcoming the challenges posed by complex inter-variable interactions and dynamic temporal patterns.

D. Qualitative Results

Since numerical metrics alone cannot fully assess the quality of imputation, we visualize the imputation results in Figs. 3 and 4 to evaluate the effectiveness of our proposed method. Fig. 3 visualizes the correlations between variables in the Weather dataset using cluster maps and dendrograms for the original data and the data imputed by Fusion-MTSI (Pearson) and Fusion-MTSI (Spearman). The first row (a) represents the cluster map and dendrogram of the original data, while the second row shows the results for imputed data: (b) corresponds to Fusion-MTSI (Pearson), and (c) to Fusion-MTSI (Spearman). The dendrogram visually depicts the clustering of variables, and the closer the color patterns in the cluster maps and the structures of the dendrograms are to those of the original data, the better the imputation method's performance.

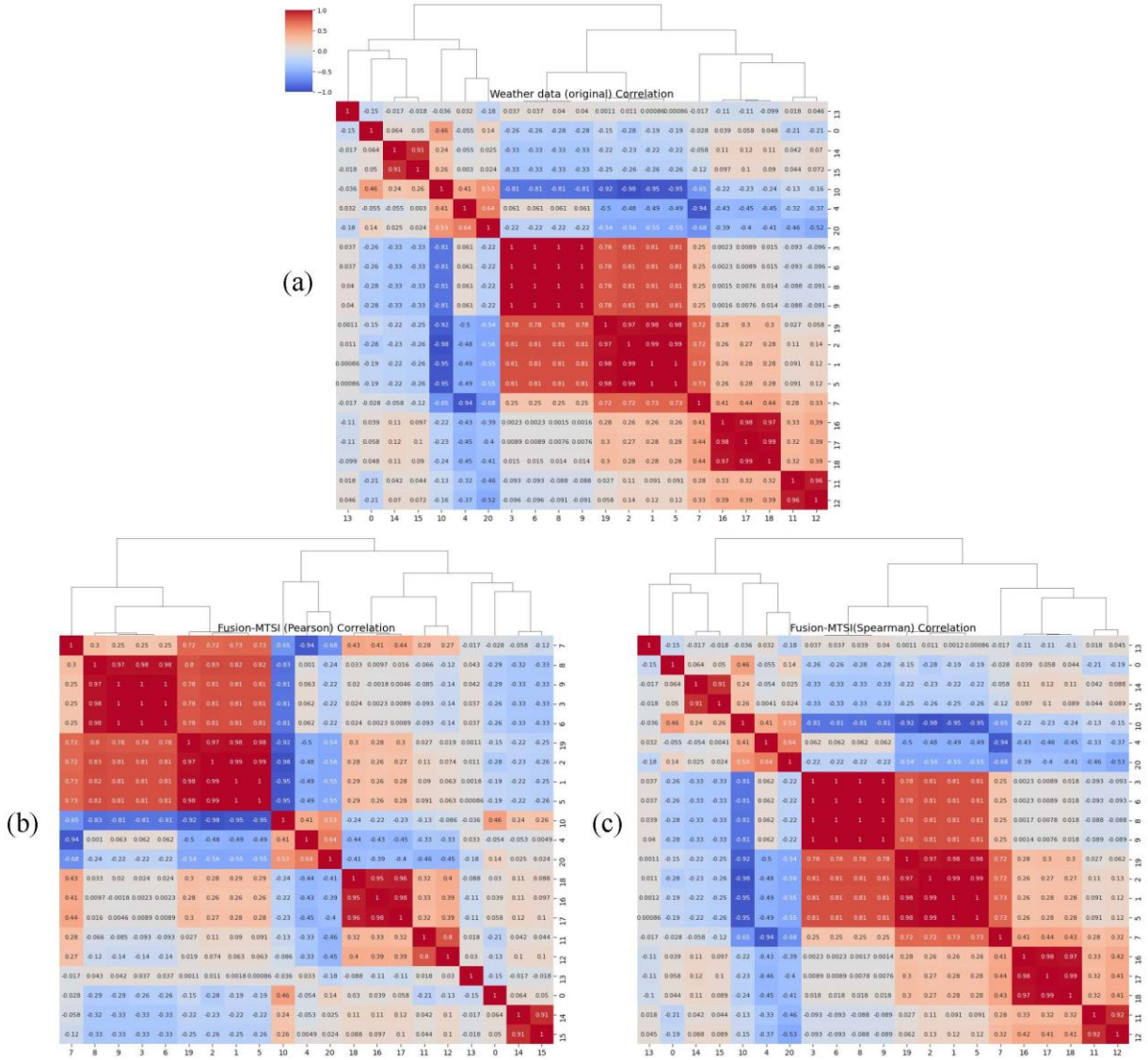

Fig. 3. Cluster Map and Dendrogram Visualization of Variable Correlations: Original vs. Imputed Data (Fusion-MTSI).

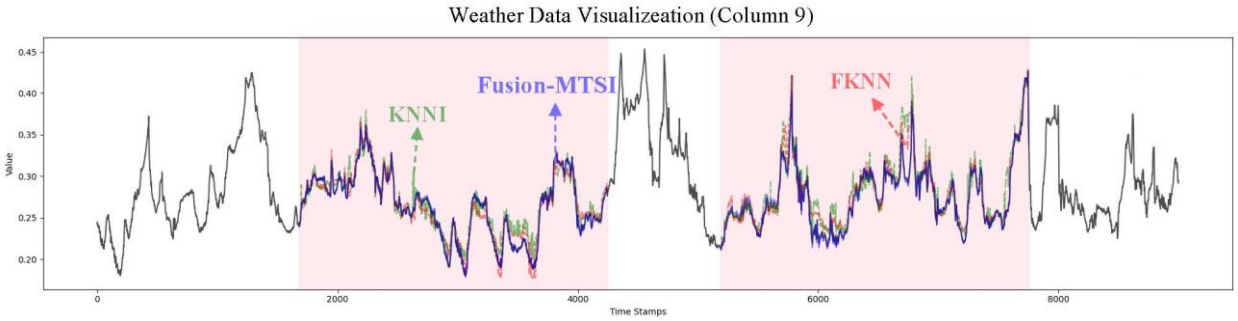

Fig. 4. Visualization of imputation in the missing region of column 9 in the Weather dataset.

Cluster Structure of the Original Data: The cluster map of the original data reveals two primary clusters.

- A cluster composed of variables: [13, 0, 14, 15, 10, 4, 20]
- A cluster composed of variables: [3, 6, 8, 9, 19, 2, 1, 5, 7, 16, 17, 18, 11, 12]

Cluster Structure of Data Imputed by Fusion-MTSI (Pearson): The data imputed by Fusion-MTSI (Pearson) formed the following clusters.

- A cluster composed of variables: [7, 8, 9, 3, 6, 19, 2, 1, 5]
- A cluster composed of variables: [10, 4, 20, 18, 16, 17, 11, 12, 13, 0, 14, 15]

The results for the Cluster Structure of Data Imputed by Fusion-MTSI (Pearson) do not perfectly match the cluster structure of the original data, with some variables being grouped differently. This discrepancy indicates that the Pearson correlation-based approach fails to fully restore the inter-variable relationships.

Cluster Structure of Data Imputed by Fusion-MTSI (Spearman): The data imputed by Fusion-MTSI (Spearman) formed the following clusters.

- A cluster composed of variables: [13, 0, 14, 15, 10, 4, 20]
- A cluster composed of variables: [3, 6, 8, 9, 19, 2, 1, 5, 7, 16, 17, 18, 11, 12]

The results for the Cluster Structure of Data Imputed by Fusion-MTSI (Spearman) demonstrates that the method using Spearman correlation effectively captures the non-linear relationships between variables and accurately restores the correlation and clustering structure of the original data. Fusion-MTSI (Spearman) outperforms Fusion-MTSI (Pearson) by effectively restoring the inter-variable correlations and cluster structures of the original data. Its ability to capture non-linear relationships demonstrates superior performance in handling complex multivariate time series data. In contrast, Fusion-MTSI (Pearson) struggles to reflect non-linear relationships, as evidenced by mismatched clusters. These findings establish Fusion-MTSI with Spearman correlation as a more robust and effective approach for the imputation of complex multivariate time series data.

Fig. 4 illustrates the imputation outcomes for column 9 of the Weather dataset, where missing values were artificially introduced to evaluate performance under controlled conditions. The black line represents the ground truth values, while the pink-shaded area highlights the regions with missing data. Imputed values are depicted using different colors: green for KNNI, red for FKNN, and blue for the proposed method. Consistent with the quantitative results presented in Table II. the proposed method (blue) shows a strong ability to closely follow the ground truth (black), demonstrating minimal error and high accuracy. FKNN (red) ranks second in performance but exhibits larger deviations from the ground truth, while KNNI (green) shows significantly higher errors, struggling to handle the complex temporal dependencies of the dataset.

V. Conclusion

In this paper, we propose Fusion-MTSI to overcome the limitations of existing imputation methods, which often struggle to handle both non-linear inter-variable interactions and contiguous missing segments in complex time series data. By combining feature-wise and point-wise comparisons, Fusion-MTSI effectively captures both global and local patterns, delivering robust performance in restoring missing values while reflecting inherent dependencies and causal relationships among variables. Notably, experiments on six real-world time series datasets demonstrate that Fusion-MTSI reduces MSE by up to 31%, showcasing its ability to excel even in complex datasets with strong seasonality. Furthermore, by minimizing reliance on domain-specific knowledge, Fusion-MTSI expands its applicability to fields where expert intervention is limited. Despite these advantages, Fusion-MTSI is not directly applicable to univariate data or discrete variables. While Fusion-MTSI removes the need for tuning window sizes, it still requires selecting both the optimal number of neighbors k and the number of similar features k_f. As a future direction, it would be beneficial to explore an adaptive feature selection technique that can dynamically adjust k and k_f based on evolving data patterns, thereby further enhancing the method's flexibility.

Appendix A: Parameter Sensitivity

A. Quantitative Results

In Table AI, we present a parameter sensitivity analysis to evaluate the performance of our proposed method, Fusion-MTSI, across six real-world datasets. This analysis examines the excellence of Fusion-MTSI under varying conditions, including changes in the number of missing values, the length of consecutive missing values, and the selection of the most relevant features as a hyperparameter. For this experiment, we varied the ratio of missing values to the total number of data points, $r_{\text{missing}} \in \{0.025, 0.05, 0.075, 0.1\}$, the maximum length of consecutive missing values, $l_{\text{missing}} \in \{0.025, 0.05, 0.075, 0.1\}$ and the number of most relevant features, $k_f \in \{3,5,7,10\}$, with default values set to $r_{\text{missing}} = 0.05$, $l_{\text{missing}} = 0.05$ and $k_f = 5$. Table AI shows that, for r_{missing}, lower ratios of missing data generally result in lower MSE values across five datasets, with Electricity as an exception. When performance improves relative to the default setting, we observe significant gains of approximately 23%, 34%, and 64% in the Exchange, Weather, and ETT* datasets, respectively. This outcome is expected as the reduction in missing values increases the availability of reference variables and data. For l_{missing}, four datasets (Electricity, Traffic, Weather, and ILI) exhibit improved performance when l_{missing} remains at or below the default setting. In contrast, Exchange and ETT* show performance gains even as l_{missing} increases, with improvements of 9% and 49%, respectively, over the default values. Finally, with respect to k_f, performance is generally better for Exchange, Weather, ILI, and ETT* when the number of reference variables is equal to or fewer than the default value. Exchange, which has a total of eight variables, achieves optimal performance when referencing five variables. Conversely, performance improves for Electricity and Traffic when k_f is set to 7 or 10. These two datasets contain more than 300 multivariate features, and thus benefit from a larger reference set, enhancing imputation performance.

TABLE AI. PARAMETER SENSITIVITY PERFORMANCE RESULTS OF THE PROPOSED METHOD

Dataset	Electricity		Exchange		Traffic		Weather		ILI		ETT*	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$r_{\text{missing}} = 0.025$	0.0033	0.0356	**0.0019**	**0.0346**	**0.0012**	0.0150	**0.0007**	**0.0110**	0.0040	0.0295	**0.0004**	**0.0135**
$r_{\text{missing}} = 0.05$	0.0030	0.0350	0.0025	0.0371	**0.0012**	**0.0148**	0.0011	0.0135	**0.0022**	**0.0220**	0.0012	0.0192
$r_{\text{missing}} = 0.075$	**0.0026**	**0.0333**	0.0035	0.0415	0.0013	0.0157	0.0010	0.0147	0.0035	0.0304	0.0075	0.0402
$r_{\text{missing}} = 0.1$	**0.0026**	0.0334	0.0038	0.0446	0.0014	0.0163	0.0033	0.0254	0.0042	0.0359	0.0037	0.0368
$l_{\text{missing}} = 0.025$	**0.0022**	**0.0317**	0.0039	0.0423	**0.0011**	**0.0141**	0.0027	0.0212	**0.0019**	0.0260	0.0033	0.0353
$l_{\text{missing}} = 0.05$	0.0030	0.0350	0.0025	0.0371	0.0012	0.0148	**0.0011**	**0.0135**	0.0022	**0.0220**	0.0012	0.0192
$l_{\text{missing}} = 0.075$	0.0028	0.0350	**0.0023**	**0.0355**	0.0016	0.0179	0.0016	0.0190	0.0080	0.0531	0.0015	0.0228
$l_{\text{missing}} = 0.1$	0.0031	0.0377	0.0080	0.0674	0.0017	0.0197	0.0026	0.0242	0.0029	0.0252	**0.0006**	**0.0146**
$k_f = 3$	0.0033	0.0370	0.0150	0.0802	0.0013	0.0156	**0.0007**	**0.0101**	**0.0017**	**0.0204**	**0.0012**	0.0203
$k_f = 5$	0.0030	0.0350	**0.0025**	**0.0371**	0.0012	0.0148	0.0011	0.0135	0.0022	0.0220	**0.0012**	**0.0192**
$k_f = 7$	0.0029	0.0340	**0.0025**	**0.0371**	**0.0011**	**0.0145**	0.0016	0.0185	0.0024	0.0222	0.0014	0.0220
$k_f = 10$	**0.0026**	**0.0324**	**0.0025**	**0.0371**	**0.0011**	**0.0145**	0.0017	0.0201	0.0024	0.0222	0.0014	0.0220

B. *Qualitative Results*

In Fig. A1, we focus on the 32,000–33,000 timestamp range of column 9 in the Weather dataset, a segment known to be challenging for imputation, and visualize the results based on different values of k_f. The imputed values are represented by dashed lines as follows: $k_f = 3$ (blue), $k_f = 5$ (orange), $k_f = 7$ (green), and $k_f = 10$ (red). Consistent with Table AI, our proposed method performs better on the Weather dataset as k_f decreases, a trend that is visually confirmed in the figure. As the value of k_f decreases, the imputed lines align more closely with the ground truth (black), while higher values of k_f result in greater deviation from the ground truth line. Notably, at $k_f = 7$, our method achieves an MSE of 0.0017, comparable to FKNN MSE of 0.0016. At $k_f = 10$, the MSE rises to 0.0017, closely matching KNNI MSE of 0.0017. Building on our parameter sensitivity analysis, the findings emphasize the importance of hyperparameter in determining imputation accuracy across varying datasets and conditions. This insight reinforces the need for adaptive mechanisms to optimize parameter selection dynamically.

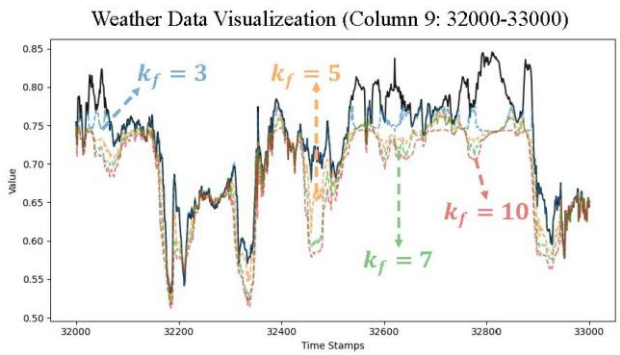

Fig. A1. Imputation in the 32,000–33,000 timestamp range of column 9 in the Weather dataset using Fusion-MTSI.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Sangyong Lee developed the research concept and study framework, conducted experiments, and wrote the article. Subo Hwang proposed improvements to the approach, reviewed the document, and contributed to writing the article. All authors had approved the final version.

FUNDING

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2022-0-00047, Development of microservices development/operation platform technology that supports application service operation intelligence).

REFERENCES

[1] M. Lepot, J.-B. Aubin, and F. H. Clemens, "Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment," *Water*, vol. 9, no. 10, 796, 2017.

[2] J.-H. Böse *et al*., "Probabilistic demand forecasting at scale," in *Proc. VLDB Endow*, Aug. 2017, vol. 10, no. 12, pp. 1694–1705. doi: 10.14778/3137765.3137775

[3] T. G. Andersen, T. Bollerslev, P. Christoffersen, and F. X. Diebold, "Volatility forecasting," National Bureau of Economic Research Cambridge, Mass., USA, 2005.

[4] C.-Y. Hsu and W.-C. Liu, "Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing," *J. Intell. Manuf.*, vol. 32, no. 3, pp. 823–836, Mar. 2021. doi: 10.1007/s10845-020-01591-0

[5] A. Gutiérrez, J. Bobadilla, and S. Alonso, "Comparison of models for predicting the number of calls received in a call center through time series analysis," *Journal of Advances in Information Technology*, vol. 15, no. 11, 2024.

[6] S. Namasudra, S. Dhamodharavadhani, R. Rathipriya, R. G. Crespo, and N. R. Moparthi, "Enhanced neural network-based univariate

time-series forecasting model for big data," *Big Data*, vol. 12, no. 2, pp. 83–99, Apr. 2024. doi: 10.1089/big.2022.0155
[7] K. Neusser, *Time Series Econometrics. in Springer Texts in Business and Economics*, Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-32862-1
[8] S. Vaughan, "Random time series in astronomy," *Phil. Trans. R. Soc. A.*, vol. 371, no. 1984, 20110549, Feb. 2013. doi: 10.1098/rsta.2011.0549
[9] A. Prokoph and F. Barthelmes, "Detection of nonstationarities in geological time series: Wavelet transform of chaotic and cyclic sequences," *Computers & Geosciences*, vol. 22, no. 10, pp. 1097–1108, 1996.
[10] B. O. Abisoye, Y. Sun, and Z. Wang, "Machine learning forecasting model for solar energy radiation," *International Journal of Computer Theory and Engineering*, vol. 16, no. 2, pp. 66–75, 2024.
[11] M. Mudelsee, "Trend analysis of climate time series: A review of methods," *Earth-Science Reviews*, vol. 190, pp. 310–322, 2019.
[12] D. S. Stoffer and H. Ombao, "Editorial: Special issue on time series analysis in the biological sciences," *Journal Time Series Analysis*, vol. 33, no. 5, pp. 701–703, Sep. 2012. doi: 10.1111/j.1467-9892.2012.00805.x
[13] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, no. 1, pp. 44–56, 2019.
[14] B. Agbo, H. Al-Aqrabi, R. Hill, and T. Alsboui, "Missing data imputation in the Internet of Things sensor networks," *Future Internet*, vol. 14, no. 5, 143, 2022.
[15] J. Kaiser, "Dealing with missing values in data," *Journal of Systems Integration*, vol. 5, no. 1, 2014.
[16] X. Yu, "Application research of spline interpolation and ARIMA in the field of stock market forecasting," arXiv preprint, arXiv:2311.10759, Nov. 14, 2023. doi: 10.48550/arXiv.2311.10759
[17] D. J. Stekhoven and P. Bühlmann, "MissForest—Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.
[18] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?" *Int. J. Methods Psych. Res.*, vol. 20, no. 1, pp. 40–49, Mar. 2011. doi: 10.1002/mpr.329
[19] O. Troyanskaya *et al*., "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.
[20] S. Oehmcke, O. Zielinski, and O. Kramer, "kNN ensembles with penalized DTW for multivariate time series imputation," in *Proc. 2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2016, pp. 2774–2781.
[21] A. Almeida, S. Brás, S. Sargento, and F. C. Pinto, "Focalize K-NN: An imputation algorithm for time series datasets," *Pattern Anal. Applic.*, vol. 27, no. 2, 39, Jun. 2024. doi: 10.1007/s10044-024-01262-3
[22] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. the 3rd International Conference on Knowledge Discovery and Data Mining*, 1994, pp. 359–370.
[23] A. Rovetta, "Raiders of the lost correlation: A guide on using Pearson and Spearman coefficients to detect hidden correlations in medical sciences," *Cureus*, vol. 12, no. 11, 2020.
[24] M. Alabadla, F. Sidi, I. Ishak, H. Ibrahim, L. S. Affendey, and H. Hamdan, "ExtraImpute: A novel machine learning method for missing data imputation," *Journal of Advances in Information Technology*, vol. 13, no. 5, pp. 470–476, 2022.
[25] J. Wang *et al*., "Deep learning for multivariate time series imputation: A survey," arXiv preprint, arXiv:2402.04059, Feb. 06, 2024. doi: 10.48550/arXiv.2402.04059
[26] H.-H. Hsu, A. C. Yang, and M.-D. Lu, "KNN-DTW based missing value imputation for microarray time series data," *J. Comput.*, vol. 6, no. 3, pp. 418–425, 2011.
[27] A. Stefan, V. Athitsos, and G. Das, "The move-split-merge metric for time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1425–1438, 2012.
[28] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor MI USA: ACM, Jun. 2018, pp. 95–104. doi: 10.1145/3209978.3210006