IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 July 2024, accepted 15 August 2024, date of publication 22 August 2024, date of current version 2 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3448236

RESEARCH ARTICLE

Anomaly Signal Imputation Using Latent Coordination Relations

THASORN CHALONGVORACHAI AND KUNTPONG WORARATPANYA, (Member, IEEE)
School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Lat Krabang, Bangkok 10520, Thailand
Corresponding author: Kuntpong Woraratpanya (kuntpong@it.kmitl.ac.th)

This work was supported by the King Mongkut's Institute of Technology Ladkrabang, under Grant KDS 2022/009.

ABSTRACT Missing data is a critical challenge in industrial data analysis, particularly during anomaly incidents caused by system equipment malfunctions or, more critically, by cyberattacks in industrial systems. It impedes effective imputation and compromises data integrity. Existing statistical and machine learning techniques struggle with heavily missing data, often failing to restore original data characteristics. To address this, we propose Anomaly Signal Imputation Using Latent Coordination Relations, a framework employing a variational autoencoder (VAE) to learn from complete data and establish a robust imputation model based on latent space coordination points. Experimental results from a water treatment testbed show significant improvements in output signal fidelity despite substantial data loss, outperforming conventional techniques.

INDEX TERMS Data imputation, time series analysis, anomaly detection, neural networks, variational autoencoder, latent coordination relations.

I. INTRODUCTION

Anomaly detection and classification in industry present significant challenges in machine learning, requiring advanced methodologies. Accurately identifying and categorizing anomalous events is crucial for preventing malicious cyberattacks that aim to disrupt machinery, mitigating potential catastrophic incidents, and safeguarding human lives.

The implementation of machine learning systems for anomaly detection and prediction in industrial scenarios presents significant complexities in real-world applications [1], [2]. These challenges arise from various obstacles, including the scarcity of data due to the rarity of events and privacy concerns regarding sensitive information. This paper primarily focuses on addressing data loss during anomaly incidents [3], which may be caused by system equipment malfunctions or, more critically, by cyberattacks in industrial systems.

One efficient technique to restore signal loss is data imputation. Traditional data imputation methods predominantly rely on statistical techniques, including: (i) Data deletion, which removes records with missing values, resulting in substantial information loss [4], [5], [6]; (ii) Imputing missing values with fixed numbers or the mean/median, potentially oversimplifying dataset variability [7], [8], [9]; and (iii) Hot-deck and cold-deck imputation, utilizing attributes from other datasets to fill missing data, thereby introducing inconsistencies and bias, especially in datasets with extensive missing data [10], [11], [12]. Despite their occasional effectiveness, these methods face challenges when applied to continuous data, particularly signal information, where maintaining temporal coherence is critical. Moreover, they can introduce significant bias by overly relying on a limited subset of records, such as anomalous datasets, thereby compromising the overall accuracy and reliability of the imputation process.

To address the limitations inherent in traditional imputation methods, researchers have introduced the K-nearest neighbors (KNN) approach for handling missing data [13], [14], [15]. This method involves identifying the nearest data points based on learned samples and utilizing their values to impute the missing data. The methodology leverages the similarity between data points to provide more accurate imputations. However, despite its potential, the KNN method has several drawbacks. It can exhibit poor performance, especially when dealing with large volumes of missing data. Additionally, there is a significant risk of introducing bias during the

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Crippa.

imputation process, which can affect the overall accuracy and reliability of the results.

Another attempt [16] proposed using machine learning for data imputation by implementing a variational autoencoder (VAE) [17]. This approach involves learning the complete data distribution through a VAE, encoding the data with missing values, reconstructing it via the decoder, and replacing the missing values with the reconstructed data. The authors demonstrated that this technique could successfully impute simulated missing windmill sensor data at various speeds. However, the method has notable drawbacks. It struggles with heavily missing data, leading to suboptimal performance, and does not fully address potential issues when applied to datasets containing anomalous samples.

Therefore, this paper proposes Latent Coordination Relations that aim to improve upon the previous VAE attempt [16] for imputing data and pushing the boundaries of performance. As for the procedures, visualized in detail in Fig. 1, this framework first learns complete data through a variational autoencoder to obtain the latent space. It then learns the coordination points of samples in the latent space via a feed-forward neural network. To impute missing data with this approach, the framework inserts partially lost signals into the trained encoder from an already-learned variational autoencoder to obtain the coordination position in the latent space. After that, it sets that coordination as an input in the learned feed-forward neural network to predict the output coordination. Then, it feeds that output into the learned decoder to reconstruct the signal. Lastly, it replaces only missing values with the data gained from this technique. With this approach, one can gain a significant advantage in both data imputation and anomaly detection tasks.

To validate our method, comprehensive experiments were conducted using the proposed framework on datasets from iTrust, a prominent cybersecurity organization. Specifically, the Water Treatment (SWaT) and Water Distribution (WADI) datasets [18], [19] were employed, as elaborated in Section IV-B. These datasets are derived from water treatment testbeds where researchers engineered cyberattacks to disrupt workflow or induce machinery breakdowns. Each dataset comprises distinct cyberattacks, exerting varying effects on sensor signals, thereby yielding both complex and simple anomaly patterns. Our investigation replicated heavy data missing scenarios with up to 90% of the total signal lost in both contexts. The findings underscore the superior performance of the proposed methods compared to baseline approaches in terms of data restoration fidelity and enhancements in machine learning performance, even when confronted with complex anomaly signals. This achievement can be attributed to the efficacy of enhanced data imputation methods that facilitate neural networks in accurately discerning anomalous events.

To summarize, the main contributions of this paper are listed as follows:

1) The use of latent space coordination relations can substantially assist in restoring and imputing heavily lost complex anomaly data to be like original signals.
2) Restored missing data using the proposed method helps improve the performance of anomaly detection since the signals have been restored back to almost original-like
3) The experiments can proves the above 1) and 2) contributions.

II. PROBLEM STATEMENT

In real-world datasets, missing data is a common issue caused by technical malfunctions during data capture or transfer, as well as unintentional errors during data management. These missing values can have significant consequences, leading to biased reports and impacting the performance of machine learning models designed for specific tasks. Ignoring missing data can distort analysis results, making findings less accurate and unreliable. It can also hinder machine learning models from learning effectively and performing well for their intended purpose. Therefore, it is crucial to develop effective strategies for handling missing data to ensure trustworthy data analysis and enable the creation of efficient and accurate machine learning algorithms.

To mitigate the issue of missing data, researchers have proposed and implemented various solutions, which are outlined as follows:

A. DATA ELIMINATION

Data elimination is one of the simplest methods used to address missing data. This approach involves removing attributes or rows of samples that contain missing values [4], [5], [6] to eliminate void attributes from the analysis, thus preventing miscalculations or misunderstandings in further statistical analyses.

While data elimination is straightforward to implement, it may not be suitable for datasets with continuous data, such as time series signals. The primary concern is that removing incomplete data can lead to datasets of unequal length, impacting subsequent analyses and modeling tasks [20], [21]. Therefore, alternative approaches are necessary to effectively handle missing values in continuous data, ensuring the dataset's integrity and usability.

B. STATISTIC IMPUTATION

Another traditional approach to handle missing data involves utilizing simple mathematical and statistical methods, such as computing the mean or median of samples [22], [23], [24]. This method offers advantages over complete data removal as it preserves the data length by replacing missing values with the global or local mean or median [7], [8], [9]. However, this approach may yield unreliable results when applied to data with high variability. Additionally, when a dataset contains a significant number of missing values, relying solely on the mean or median as replacements can be inappropriate. These summary statistics may not accurately represent the missing data, leading to considerable bias during the data

analysis process. As a result, such alternative techniques are necessary to effectively address missing values, particularly in cases where data exhibits high variance or when there is a substantial amount of missingness in the dataset.

C. HOT DECK AND COLD DECK IMPUTATION

Alternative approaches commonly utilized for completing missing data include Hot Deck and Cold Deck imputations [10], [11], [12]. In the Hot Deck method, missing values are imputed by randomly selecting a value from a similar attribute within the same record. Conversely, the Cold Deck approach involves selecting data from another dataset or a previously available record as a reference to fill in the missing values. However, it is important to note that these techniques can introduce bias into the data, potentially leading to incorrect analysis, particularly in cases where the data has a high degree of missingness. In such scenarios, the availability of suitable donors becomes limited, resulting in a lack of diversity in the imputed values. Consequently, the imputation process may yield the same or similar value for all missing values, further aggravating the potential for biased results.

D. K-NEAREST NEIGHBOR IMPUTATION

An advanced and increasingly popular approach to tackle the missing data problem is through the application of machine learning mechanisms, particularly the K-Nearest Neighbor (KNN) imputation technique. KNN imputation harnesses the power of machine learning algorithms to address missing values by learning from existing data points that remain unaffected by the missing data issue. The algorithm learns the attributes and characteristics of complete instances and utilizes this knowledge to identify the nearest neighbors based on a user-defined K-value [13], [14], [25]. Subsequently, the missing values are replaced with numbers obtained from these nearest neighbors. The KNN imputation method has gained attention as an effective approach for data imputation due to its capability to capture underlying patterns and relationships in the dataset [26], [15], [27]. However, like any technique, it has certain limitations. One limitation is that KNN imputation heavily relies on the available learned data and may encounter challenges when dealing with a limited amount of samples, such as anomalous incident data that rarely happens in real-world scenarios. Furthermore, the technique's weakest point lies in its performance with noisy data, such as anomalous signals, as the presence of noise can affect the accuracy of the imputed values. Despite these limitations, KNN imputation represents a valuable and promising approach in the field of missing data imputation, warranting further research and refinement.

E. DEEP LEARNING TECHNIQUES FOR DATA IMPUTATION

To address complexities within high-dimensional data, numerous researchers have introduced frameworks and methodologies rooted in deep learning paradigms. Notably, Recurrent Neural Networks (RNNs) and transfer learning techniques have been employed to impute scientific data, such as the monthly frequency of sunspots [28]. Similarly, Long Short-Term Memory Neural Networks (LSTMs) have shown promise in restoring vehicle speed data [29]. Another significant approach involves the Denoising Deep Belief Network architecture, which excels in denoising, imputation, and dimensionality reduction for industrial data [30]. Furthermore, various deep learning techniques have been adapted and modified for data imputation across different applications, demonstrating their versatility and effectiveness [31], [32], [33]. Despite the advances, these methodologies require continuous refinement to enhance their robustness and applicability in handling diverse datasets.

While these state-of-the-art techniques hold promise, their effectiveness is notably impacted by the scarcity of anomalous data. Deep learning inherently demands a substantial volume of data to achieve optimal performance. Insufficient samples during the training phase often lead to an overfitted model, resulting in imprecise imputation outputs. Consequently, the applicability of this approach to address anomaly data imputation tasks, as discussed in this paper, is notably unsuitable for handling such challenges.

F. VARIATIONAL AUTOENCODER FOR DATA IMPUTATION

The Variational Autoencoder (VAE) is a neural network architecture introduced in [17]. In essence, the fundamental concept of this approach revolves around implementing the VAE model, which comprises an encoder and decoder within the neural network. By utilizing the mean and standard deviation vectors, the VAE generates a probability distribution in the latent space. Mathematically, the VAE is defined by (1).

$$\begin{aligned} &\log P(X) - D_{KL}[Q(Z|X)||P(Z|X)] \\ &\qquad = E[\log P(X|Z)] - D_{KL}[Q(Z|X)||P(Z)] \end{aligned} \tag{1}$$

where the training data are denoted by X and the latent variable is represented by Z. The conditional probability of the encoder is denoted as P, while the conditional probability of the decoder is denoted as Q. The Kullback-Leibler divergence is symbolized as D_{KL}.

VAE has demonstrated its versatility across various applications, with a particular focus on generative tasks like image generation [34], [35], [36] and image super-resolution [37], [38], [39]. However, its utility extends beyond data generation, as it excels as a reliable tool for data reconstruction. This characteristic makes it highly valuable in signal restoration applications, where it plays a crucial role in filling the missing values with reconstructed data.

In 2018, an exemplary study conducted by John et al. [16] shed light on the advantageous capabilities of VAE for data imputation. Specifically, they explored the application of VAE architecture in reconstructing and imputing missing data in a simulated milling circuit, as explained in Section III.

While our study shows advantages over traditional methods, it is important to acknowledge its limitations, especially

when dealing with heavily missing data. Because there is not enough reference data during the encoding process, the VAE's performance suffers, leading to suboptimal results during the decoder process. Additionally, the full potential of the learned latent space remains untapped. This latent space has the potential to represent data in a way that could improve data restoration. Therefore, developing a model that effectively uses the latent space could greatly enhance the process of data reconstruction and imputation. Leveraging the latent space would lead to improvements in data imputation

III. PROPOSED METHOD

To address the aforementioned challenges and fully exploit the potential of the latent space, this section introduces a novel framework called Anomaly Signal Imputation Using Latent Coordination Relations. The framework adopted the core idea from the mentioned approach in Section II.

The methodology devised by John et al [16] involved a step-by-step procedure, beginning with the training of the VAE model on clean data that contain no missing values. Subsequently, as a pre-processing step, the missing values were imputed using either a constant number or random values. The encoded representation of the data was then obtained using an encoder, which transformed the input data into latent variables. The decoder component of the VAE model played a crucial role in the reconstruction process by transforming the latent variables back into the original data. Importantly, this method replaced the missing values with the reconstructed data while preserving the integrity of the original data.

To simplify the data restoration procedures, the following steps can be summarized:

1) Replace every missing data and Not a Number (NaN) value with 0 or a random number;
2) Feed the pre-processing data into a trained encoder. At this step, it will yield latent variables;
3) Feed the latent variables into the trained decoder. In this process, it will yield the reconstructed data; and
4) Substitute every reconstructed sample at the same position as the original input that contains missing data or a NaN value, while the initial data remains stable.

While this method may seem prominent for the data imputation task, it still faces challenges in scenarios involving heavily lost data. Moreover, there is room for improvement to leverage the latent space for further advantages.

Therefore, to overcome the limitations of existing approaches and fully utilize the latent space, this paper proposes a framework that uses latent coordinates obtained from the VAE model and applies neural networks to learn and predict these coordinates as closely as possible to the original ones in the latent space. This enables the decoder to accurately reconstruct the signal.

The overall concept of the framework is depicted in Fig. 1. Below, we outline the step-by-step description of the framework.

A. TRAINING PHASE

The training phase aims to achieve several key objectives, including acquiring the encoder and decoder models from the VAE latent space and training a prediction model for latent coordinate points using a feed-forward neural network. The procedures involved in this phase are outlined below:

1) DATA PREPROCESSING

Prior to training, it is important to note that the data used in this phase is complete data. The input data undergoes preprocessing steps such as normalization and data augmentation to ensure compatibility with the subsequent models.

2) VAE TRAINING

The VAE model is trained using the entire dataset, as illustrated in the VAE Training module in Fig. 1(a). Each sensor signal is independently trained with its corresponding VAE, rather than employing a single VAE to collectively process all sensor signals. The encoder component of the VAE maps the input data to a lower-dimensional latent space, while the decoder reconstructs the input data from the latent space. During this step, the focus lies on obtaining the latent space and coordination points that accurately represent the complete samples learned through the VAE. Comprehensively, latent space can be defined as stated in *Definition 1*:

Definition 1 (Latent Space): The Latent Space Z is a set of real numbers that consist of coordination point z, where $z \in Z; Z \in \mathbb{R}$.

Note that Z from Eq. (1) and *Definition 1* refers to the same entity: the set of latent variables obtained after training the VAE model.

3) LATENT COORDINATION RELATION TRAINING

This stage forms the main concept of this paper, as depicted in the Latent Coordination Relation Training procedure in Fig. 1(a). A feed-forward neural network is constructed to predict latent coordinate points obtained from the VAE encoder, which serve as inputs, while the corresponding target coordinate points act as ground truth labels. The objective of this prediction model is to establish correlations between the latent coordinates of the sensor data, enabling latent coordination of the sensors to aid in predicting and restoring signal data in subsequent steps.

4) MODEL EXTRACTION

Once the VAE and latent relation prediction models are trained, the encoder and decoder models are extracted from the VAE architecture, along with the latent relation prediction model. The encoder model facilitates the transformation of input data into the latent space, while the decoder model reconstructs data from the latent space back to the original input space. The latent coordination model, on the other hand, enables the discovery of relationships between each sensor's data, aiding in the restoration of missing or incomplete data.

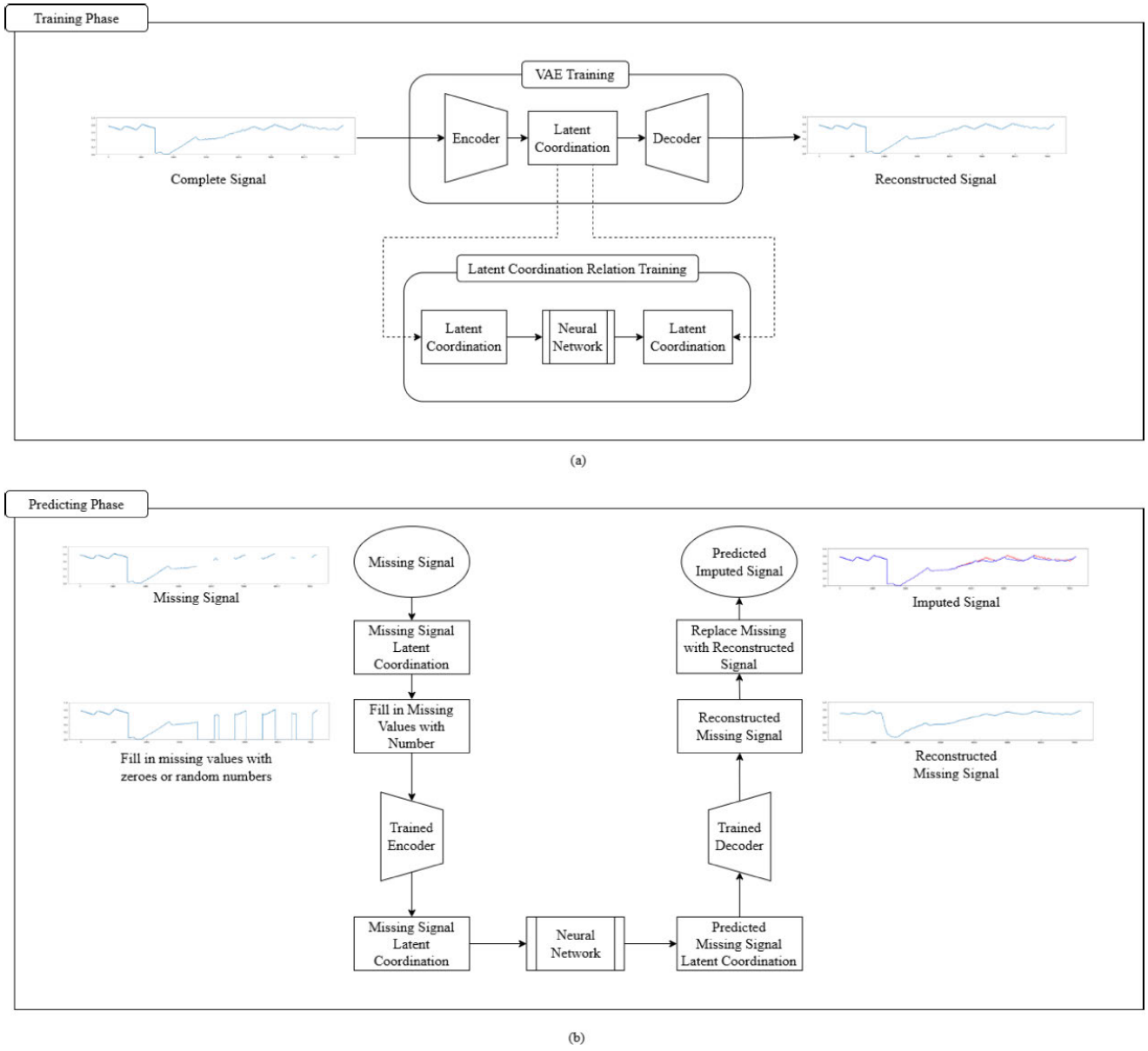

FIGURE 1. The overall framework of the Anomaly Signal Imputation Using Latent Coordination Relations: (a) is training phase and (b) is predicting phase.

By following these outlined procedures, the ultimate goal of obtaining the encoder and decoder models from the VAE latent space, as well as the prediction model for latent coordinate points, can be accomplished.

B. IMPUTATION PHASE

The imputation phase utilizes the encoder, decoder, and latent relation prediction models obtained from the training phase for data imputation. The steps involved in this phase are as follows:

1) DATA PREPROCESSING

Similar to the training phase, the data undergoes preprocessing steps to handle missing values. In this case, the data includes both the samples with missing values and those with complete values. Additionally, the missing values (NaN values) within the dataset are addressed by using two distinct approaches: filling with zeroes and imputing with randomly generated numbers. These methods are applied separately for each imputation case, ensuring that missing values are appropriately handled before inputting the data into the VAE model. This ensures that the dataset is complete and ready for further processing and analysis within the VAE framework.

2) ENCODING THE MISSING DATA

During this stage, the framework processes the input signal with missing values by feeding it into the encoder component of the VAE, as established in the previous phase. This

procedure is illustrated in Fig. 1(b), where the trained encoder component is depicted. By mapping the input signal with missing values to the latent space, the framework generates the latent coordinates based on the learned data from the training phase. This step helps identify the closest data points in the latent space that are similar to the missing data point, allowing for efficient imputation. The encoder in this stage can be defined as *Definition 2*.

Definition 2 (Encoder): Encoder E is a function that encodes a sample d to a coordination point z on latent space Z, such that $E: d \rightarrow z; z \in \mathbb{R}$ and $z \in Z$.

3) LATENT PREDICTION

In this step, a neural network is utilized to predict the relationships between the latent coordinates projected onto the latent space, as illustrated in Fig. 1(b). This model is derived from the training phase, where it learned the latent coordination from the training samples. The concept of the latent prediction function can be expressed in the *Definition 3*.

Definition 3 (Latent Coordination Relation): The Latent Coordination Relation is the function f that maps the coordination point of Latent Space Z_a with another latent space Z_n, $f : Z_a \rightarrow Z_n$. Therefore, coordination point in Z_a and Z_n are also mapped through $f : z_a \rightarrow z_n$, where z_a, $z_n \in \mathbb{R}$, $z_a \in Z_a$, and $z_n \in Z_n$.

It is important to note that Z_a and Z_n are latent spaces derived from different VAE trainings and models. Therefore, these variables are distinct, and the latent coordinate relation function (f) serves as a tool to establish connections between these latent spaces.

This prediction model is to identify and capture the underlying relationships of its own sensor signal. By seeking these relationships through this function (*Definition 3*), the model assists in predicting and restoring the missing data in the subsequent steps of the imputation process.

4) DATA RECONSTRUCTION WITH THE DECODER

Once the latent coordinates from the model are obtained, they are used as input for the decoder. The decoder acts as a reconstruction tool within the VAE framework, restoring the shape of the signal based on the latent coordinates projected by the encoder. This step involves reconstructing the missing data and results in a signal that restores most of the lost data, improving the completeness of the dataset. Mathematically, the idea of the decoder is stated below:

Definition 4 (Decoder): Decoder D is a function that reconstructs a coordination point z as close to the original data $\hat{d}$ as possible, hence, $D: x \rightarrow \hat{d}; x \in \mathbb{R}$ and $z \in Z$.

To clarify, d from *Definition 3* represents the original sample, while $\hat{d}$ denotes the reconstructed sample aimed to closely resemble d.

5) DATA IMPUTATION

This imputation step effectively fills in the missing values, ensuring a more complete dataset for further analysis, training, or implementation in reliable machine learning models. This process is equivalent to the replace missing values with the reconstructed signal procedure as depicted in Fig. 1(b). Once these steps are completed, the data imputation process is finalized and the imputed dataset is ready for training, analysis, or implementation within a robust machine learning framework.

IV. EXPERIMENTAL DESIGN

This section of the paper serves the purpose of presenting a thorough performance comparison of various methods, providing a description of the dataset, explaining the process of simulating missing values, and outlining the experimental setup. The specific details are as follows:

A. PERFORMANCE COMPARISON METHODS

To conduct an experimental comparison in data imputation, two key factors must be considered: the baseline methods to be compared with the proposed method and the evaluation metrics used to assess the performance of the imputed signal, as explained below.

1) BASELINE METHODS

For performance comparison, four different baseline methods were employed: (i) Mean imputation, which involves calculating the mean based on available values [7], [8], [9]; (ii) Hot Deck imputation, wherein values from the most recent record remaining in the dataset are used to impute missing values [11], [12], [40]; (iii) KNN imputation, where the completed data is trained beforehand to identify the nearest data points for imputing missing values [13], [14]; and (iv) Variational Autoencoder for Imputation (VAE), utilizing sensor signal as inputs for a VAE model. This VAE model was trained using data from sensors within the system where anomalies occurred [16].

2) EVALUATION METRICS

The effectiveness of the proposed imputation methods was assessed using two key groups of metrics: (i) Root Mean Squared Error (RMSE) [41], [42] and Normalized Dynamic Time Warping (N-DTW) [43], [44] to evaluate signal fidelity and shape similarity, ensuring that the imputed signals retain the characteristics of the originals; and (ii) Accuracy, Precision, Recall, and F1-Score for anomaly classification, confirming the practical viability of the imputed data for real-world applications [45], [46].

The first metric employed was Root Mean Square Error (RMSE), which measures the dissimilarity between the ground truth signals and the imputed results [41]. Lower RMSE scores indicate higher similarity and superior imputation performance. The RMSE is calculated using Eq. (2):

$$r(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}{n}} \tag{2}$$

where r is the RMSE score, y is original time series, $\hat{y}$ is the imputed time series, and n is the quantity of samples.

In summary, the primary purpose of RMSE is to compare the original and imputed signals point by point.

Additionally, Normalized Dynamic Time Warping (N-DTW) serves as an additional metric, alongside RMSE, to ensure the quality of imputed signal characteristics. N-DTW normalizes the dynamic time warping value for comparison purposes, providing a comprehensive assessment of signal similarity. It is calculated by applying min-max normalization to the range between 0 and 1 and using fast dynamic time warping [43]. The N-DTW equation is expressed in Eq. (3):

$$N_d(y, \hat{y}) = \frac{D(y, \hat{y}) - \sigma_{\min}}{\sigma_{\max} - \sigma_{\min}} \tag{3}$$

where $N_d(y, \hat{y})$ is the normalized dynamic time warping, σ_{max} is the maximum value of the dynamic time warping algorithm, σ_{min} is the minimum value of the dynamic time warping algorithm, and $D(y, \hat{y})$ dynamic time warping function that calculates the distance between the actual observation time series y and the comparison time series $\hat{y}$.

The main property of N-DTW is its ability to evaluate the similarity of signal characteristics even if the imputed signal is slightly shifted. This feature is crucial in classification, as in real-world applications, the shape and characteristics of an anomaly are more important for accurate anomaly prediction than its exact position.

The last metric employed in this experiment is the standard confusion metric, consisting of: Accuracy, Precision, Recall, and F1-Score [45], which assesses the imputation models. A neural network was trained using the imputed data from each model for the task of anomalous signal classification. The accuracy score was then computed to evaluate the performance of the classification model. Higher accuracy scores indicate that the restoration method successfully restores data that closely resembles the original time series, without adversely affecting the classification model's performance. The equations of the Accuracy (S_a), Precision (S_p), Recall (S_r), and F1-Score (S_f) are denoted in Eqs. (4)-(7), respectively:

$$S_a = \frac{C_{TP} + C_{TN}}{C_{TP} + C_{TN} + C_{FP} + C_{FN}} \tag{4}$$

where C_{TP} is the true positive value, C_{TN} is the true negative value, C_{FP} is the false positive value, and C_{FN} is the false negative value.

$$S_p = \frac{C_{TP}}{C_{TP} + C_{FP}} \tag{5}$$

$$S_r = \frac{C_{TP}}{C_{TP} + C_{FN}} \tag{6}$$

$$S_f = \frac{2 \times S_p \times S_r}{S_p + S_r} \tag{7}$$

To ensure the reliability of the experimental results, a 10-fold cross-validation method was employed. This approach mitigates potential biases and variability by dividing the data into ten subsets or folds. The models were trained and evaluated on different combinations of training and testing folds, enhancing the robustness and generalizability of the conclusions drawn from the experiments.

In summary, RMSE and N-DTW serve as statistical metrics for assessing signal similarity to the original, while Accuracy, Precision, Recall, and F1-Score gauge its suitability for real-world applications.

B. DATASET EXPLANATION

To evaluate the effectiveness of the VAE model in imputing lost values, we utilized the Secure Water Treatment (SWaT) and Water Distribution (WADI) datasets from iTrust, Centre for Research in Cyber Security [18], [19]. iTrust is a cybersecurity organization established by the Singapore University of Technology and Design and the Singapore Ministry of Defence. These datasets consist of signals collected from sensors installed on electronic components within a water treatment testbed. The data captures both normal operating conditions and anomalous events, which were simulated by researchers to mimic cyberattacks. These cyberattacks aim to disrupt the system, partially damage machinery, or, in the worst case, compromise the entire testbed. The goal of this testbed is to create a dataset that closely approximates real-world scenarios, enabling researchers to study and develop preventative measures against cyberattacks.

The selection of these datasets for the experimental investigation in this study is motivated by two key factors. Firstly, they are derived from real-world systems, demonstrating their practical applicability in real-world scenarios. Secondly, these datasets are well-suited for assessing their effectiveness in data imputation, particularly when dealing with various patterns of anomalous signals.

As for the type of anomaly signals, in the testing phase, two sensor signals were specifically chosen to evaluate the performance of the VAE model in imputing lost values caused by cyber attacks: (i) The water reservoir sensor (LIT002) was targeted with a cyber attack that involved turning off valves related to the reservoir, resulting in its drainage; and (ii) The reverse osmosis water tank sensor (LIT401) experienced a cyber attack that manipulated the water level, setting it at a constant value of 1,000 mm.

These two sensors not only exhibit distinct characteristics but also display different variances in the signal, influenced by anomalies. Specifically, the LIT002 sensor tends to demonstrate higher variance compared to the LIT401 signal. This variance difference is attributed to the anomaly events causing significant fluctuations in sample values and a dramatic change in signal characteristics throughout the anomaly incident, as depicted in Fig. 2. Consequently, it is evident that LIT002 represents complex anomaly samples, while LIT401 represents simple anomalies.

These selected sensor signals with anomalous events serve as test cases to evaluate the VAE model's ability to impute missing values and restore the signals to their original patterns.

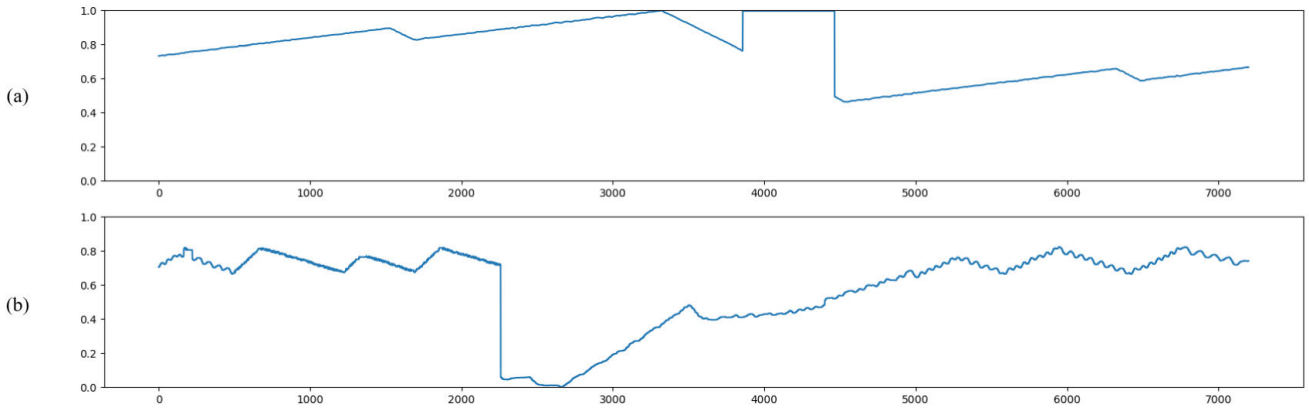

FIGURE 2. Examples from the SWaT and WADI datasets illustrating distinct anomaly scenarios. (a) depicts the behavior of the LIT401 signal under the influence of a simple attack, while (b) showcases the response of the LIT002 signal to a more complex anomalous incident.

C. MISSING DATA TYPES AND SIMULATION

Various types of missing data can occur in time series signals, particularly in electronic components such as IoT devices or machinery sensors. The literatures [16], [47], [48], [49] highlight two common types: point missing and range missing. Point missing refers to sporadic occurrences of empty values in a dataset, without any discernible pattern. These missing values are randomly scattered throughout the dataset, resulting in irregular gaps in the time series. On the other hand, range missing exhibits a specific pattern where a contiguous segment of data is missing for a certain length. This type of missing data often occurs in a structured manner, attributed to specific events or conditions.

To address these different types of missing data and provide a comprehensive evaluation, the experiment phase incorporates simulated missing data based on the mentioned patterns. The NaN values are randomly inserted into the selected sensors' signals using the following two main types: (i) Similar to the approach described in [16] and [47], two missing data scenarios are simulated: 20% of missing values (1,440 seconds in total) and 90% of missing values (6,480 seconds in total). Additionally, this experiment also include 50% of missing values cases to observe the affect of missing between minimum and maximum losses. These missing values are distributed randomly across the signals by applying Gaussian Noise; (ii) The range missing scenario from [48] is also considered. Five missing ranges are implemented per signal sample, where each missing range spans 288, 720, and 1,296 seconds, resulting in a total of 1,440, 3,600, and 6,480 seconds of missing data, respectively. The reason behind these numbers is to mimic the percentage setting from Gaussian cases.

In this paper, heavy loss cases are defined as instances where data loss amounts to 90% of the total signal, while medium loss refers to 50% loss, and low loss corresponds to 20% data loss.

D. EXPERIMENTAL SETUP

To ensure an unbiased evaluation of the VAE performance for data imputation, the experiment settings for both VAE models were standardized as follows:

- Both VAE models were configured with an input layer of 72 nodes and a second layer of 36 nodes for the encoder. The decoder architecture was set differently. The VAE models were trained using two-dimensional samples and underwent 100 epochs.
- The dataset consisted of 2,000 samples per sensor. Of these, 50% of the samples were allocated for training, while the remaining 50% were reserved for testing purposes.
- Each sample in the dataset contained 7,200 seconds of records for an anomalous signal, as described in Section IV-B.
- All data underwent normalization using the Min-Max Scaling method, which ensured that the normalized data values fell within the range of 0 and 1.
- For evaluation purposes, three main types of missing data were simulated: Gaussian Noise and Range. Within each type, varying levels of missing data were considered: 20%, 50%, and 90% to simulate low, medium, and heavy loss of signals.

Furthermore, to demonstrate the significance of imputed data in improving classification models for anomalous time series, experiments were conducted using a neural network. The experiment settings were configured as follows:

- The experiment utilizes the dataset detailed in Section IV-B. Two sensors, one demonstrating complex anomaly and the other simple anomaly characteristic, are deliberately selected to examine the imputation method's capability in managing such scenarios.
- The classification dataset consisted of a total of 2,000 samples, with 1,000 samples representing normal signals and the other half representing anomalous signals.

- 50% of the total samples was selected as a training set, while the remaining samples were allocated to the testing set for performance evaluation of the classification models.
- In the training set, the normal data were collected directly from the original sources, while the abnormal data were obtained from the imputed data generated by each imputation method.
- Similar to the VAE model, the data underwent min-max normalization, scaling the values to the range of 0-1.
- The neural network model for abnormal signal classification consisted of three layers, each with 72 nodes, and an output layer with a single node. A dropout rate of 0.5 was applied between each layer to mitigate the risk of overfitting. The neural network was trained for 100 epochs.

To gain a better understanding of the experimental procedures, we summarize them as follows:

1) Preprocess all sensor signals for training.
2) Train the VAE model with complete signals.
3) Train the Neural Network with the latent coordinates obtained from the VAE. Remarkably, both the input and output for this neural network are the same latent coordinates.
4) Preprocess testing signals by cleaning the data and addressing any missing values, either by filling them with zeros or random values.
5) Feed the test signal as an input to the trained VAE's encoder.
6) Use the previously derived latent coordinates as the input for a dedicated neural network, which predicts the expected latent coordinates based on patterns in their latent coordination relation.
7) Feed the predicted latent coordinates into the decoder component of the trained VAE to reconstruct the original signal, ensuring the data remains coherent.
8) Replace missing values with the reconstructed data.

Following the completion of these experiment steps, a rigorous evaluation is conducted, employing neural network anomaly classification models to verify the effectiveness of imputed data in supporting anomaly classification tasks. Furthermore, we apply Root Mean Square Error (RMSE) and Normalized Dynamic Time Warping (N-DTW) to assess the extent to which the reconstructed and imputed signals retain the essential characteristics of the original data. This methodology ensures the robustness and reliability of our approach in handling missing data, providing valuable insights into anomaly detection and signal integrity.

V. RESULTS AND DISCUSSION

The experimental results of each sensor's data imputation are reported in Tables 1-12, and the example of the output imputed signal is visualized in Fig. 3. Noted that additional experimental results are shown in Appendix. Additionally, each table highlights the superior method for imputing each missing data type and fill strategy.

As anticipated, our proposed methodology excels in handling complex and heavily lossy signal scenarios, particularly with range missing data around 90%. The fidelity of the imputed signal surpasses the baselines, as shown in Tables 1, 3, 7, and 8. This is evidenced by superior Root Mean Square Error (RMSE) scores, achieving 0.0145 ± 0.0040 for the LIT002 dataset (Table 1) and 0.0128 ± 0.0015 for the LIT401 dataset (Table 7). Similarly, Normalized Dynamic Time Warping (N-DTW) scores are impressive, with 8.1688 ± 0.8048 for LIT002 and 10.5954 ± 0.7159 for LIT401. Our model consistently outperforms baselines, with accuracy, precision, recall, and F1-scores all achieving 100.0000 ± 0.0000 in the LIT002 and LIT401 scenarios. This superiority is especially notable in cases with significant data loss.

In scenarios with simple anomalies and low to medium data loss, the performance disparity among various imputation methods is minimal, as most methods handle these cases effectively. For instance, in the LIT401 experiment referenced in Tables 7 and 8, with 20% Gaussian missing data, the hot deck method achieved an RMSE of 0.0000 ± 0.0000 and N-DTW of 0.1946 ± 0.0070, slightly outperforming our proposed method, which had an RMSE of 0.0003 ± 0.0001 and N-DTW of 0.1537 ± 0.3392 in the best cases. Therefore, it can be concluded that our proposed method excels in more challenging scenarios, such as those with high variance and significant data loss.

In certain scenarios, achieving flawless accuracy, precision, recall, and F1-score metrics can be attributed to the binary classification nature of the task. Despite potential inadequacies in the imputed training data, distinctive anomalies that deviate from the norm are readily recognized and categorized as abnormal instances, thus facilitating accurate classification. For instance, as shown in Table 2, the accuracy scores of our proposed method consistently reach 100.0000.

However, a comparison of results reveals that accuracy performance may decline in cases of severe signal loss, such as a 90% missing data range in the LIT002 dataset. The accuracy of the Mean Method is 50.0050 ± 0.0150, the Hot Deck Method achieves 93.8600 ± 2.1032, and K-Means scores 92.7150 ± 3.3765. The VAE method reaches a perfect score of 100.0 only in the mean fill type case. In contrast, our method consistently achieves a perfect score of 100.0 across all fill types. This strongly indicates that our method is particularly effective in reconstructing heavily lost signals.

Although the results may not deviate drastically from perfection, slight differences in accuracy, precision, recall, and F1-score are observed in certain cases. Considering these factors alongside RMSE and N-DTW metrics, it becomes evident that our method performs exceptionally well, particularly in scenarios with substantial loss of range information. As shown in Tables 1, 3, 7 and 8, the LIT401 experiment with 90% of range missing data demonstrates the following RMSE scores (Table 7): 0.0250 ± 0.0001 (Mean), 0.0229 ± 0.0009 (Hot Deck), 0.0409 ± 0.0022 (KNN), and 0.0196 ± 0.0015 (VAE). Our proposed method outperforms these with an

TABLE 1. A comparison of the RMSE results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.0098 ± 0.0003	0.0020 ± 0.0002	0.0071 ± 0.0001	—	—
	Zero	—	—	—	0.0020 ± 0.0004	0.0016 ± 0.0005
	Random	—	—	—	$\mathbf{0.0010 \pm 0.0001}$	0.0010 ± 0.0002
	Mean	—	—	—	0.0013 ± 0.0002	0.0013 ± 0.0003
50 % Missing (range)	NaN	0.0251 ± 0.0008	0.0152 ± 0.0008	0.0259 ± 0.0044	—	—
	Zero	—	—	—	0.0106 ± 0.0018	0.0081 ± 0.0014
	Random	—	—	—	0.0062 ± 0.0012	0.0052 ± 0.0011
	Mean	—	—	—	0.0079 ± 0.0009	$\mathbf{0.0049 \pm 0.0014}$
90 % Missing (range)	NaN	0.0474 ± 0.0011	0.0440 ± 0.0030	0.0375 ± 0.0003	—	—
	Zero	—	—	—	0.0175 ± 0.0014	0.0095 ± 0.0022
	Random	—	—	—	0.0370 ± 0.0066	$\mathbf{0.0145 \pm 0.0040}$
	Mean	—	—	—	0.0259 ± 0.0039	0.0157 ± 0.0044
20 % Missing (Gaussian)	NaN	0.0098 ± 0.0002	$\mathbf{0.0000 \pm 0.0000}$	0.0062 ± 0.0001	—	—
	Zero	—	—	—	0.0255 ± 0.0046	0.0164 ± 0.0058
	Random	—	—	—	0.0005 ± 0.0002	0.0004 ± 0.0001
	Mean	—	—	—	0.0004 ± 0.0001	0.0004 ± 0.0001
50 % Missing (Gaussian)	NaN	0.0245 ± 0.0004	$\mathbf{0.0001 \pm 0.0000}$	0.0155 ± 0.0001	—	—
	Zero	—	—	—	0.0042 ± 0.0016	0.0052 ± 0.0015
	Random	—	—	—	0.0051 ± 0.0016	0.0047 ± 0.0022
	Mean	—	—	—	0.0049 ± 0.0017	0.0050 ± 0.0031
90 % Missing (Gaussian)	NaN	0.0440 ± 0.0009	$\mathbf{0.0005 \pm 0.0000}$	0.0279 ± 0.0002	—	—
	Zero	—	—	—	0.0238 ± 0.0049	0.0157 ± 0.0047
	Random	—	—	—	0.0197 ± 0.0038	0.0147 ± 0.0042
	Mean	—	—	—	0.0228 ± 0.0044	0.0157 ± 0.0064

TABLE 2. A comparison of the ACCURACY results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	98.9750 ± 1.5766	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	99.9750 ± 0.0461	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	99.9950 ± 0.0150	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	99.9750 ± 0.0461	$\mathbf{100.0000 \pm 0.0000}$
50 % Missing (range)	NaN	85.9700 ± 4.2621	99.5250 ± 0.6705	95.7850 ± 1.1265	—	—
	Zero	—	—	—	99.7650 ± 0.2470	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	99.8700 ± 0.2293	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	99.5850 ± 0.4990	99.1700 ± 2.4900
90 % Missing (range)	NaN	50.0050 ± 0.0150	93.8600 ± 2.1032	92.7150 ± 3.3765	—	—
	Zero	—	—	—	99.9900 ± 0.0300	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	99.9700 ± 0.0900	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	99.8000 ± 0.6000	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	77.5750 ± 15.5481	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	50.0000 ± 0.0000	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$

RMSE of 0.0128 ± 0.0015 (Table 7). Similarly, for N-DTW scores (Table 8), the results are 15.8007 ± 0.0402 (Mean), 14.5425 ± 0.3347 (Hot Deck), 18.7235 ± 0.7288 (KNN), and 13.5004 ± 0.4600 (VAE), while our method achieves a superior score of 10.5954 ± 0.7159.

In most instances involving Gaussian noise, the Hot-Deck method may outperform the proposed approach. For example, in the LIT401 experiment with 50% Gaussian missing data, the RMSE score for the Hot-Deck method is 0.0000 ± 0.0001, while the proposed method scores

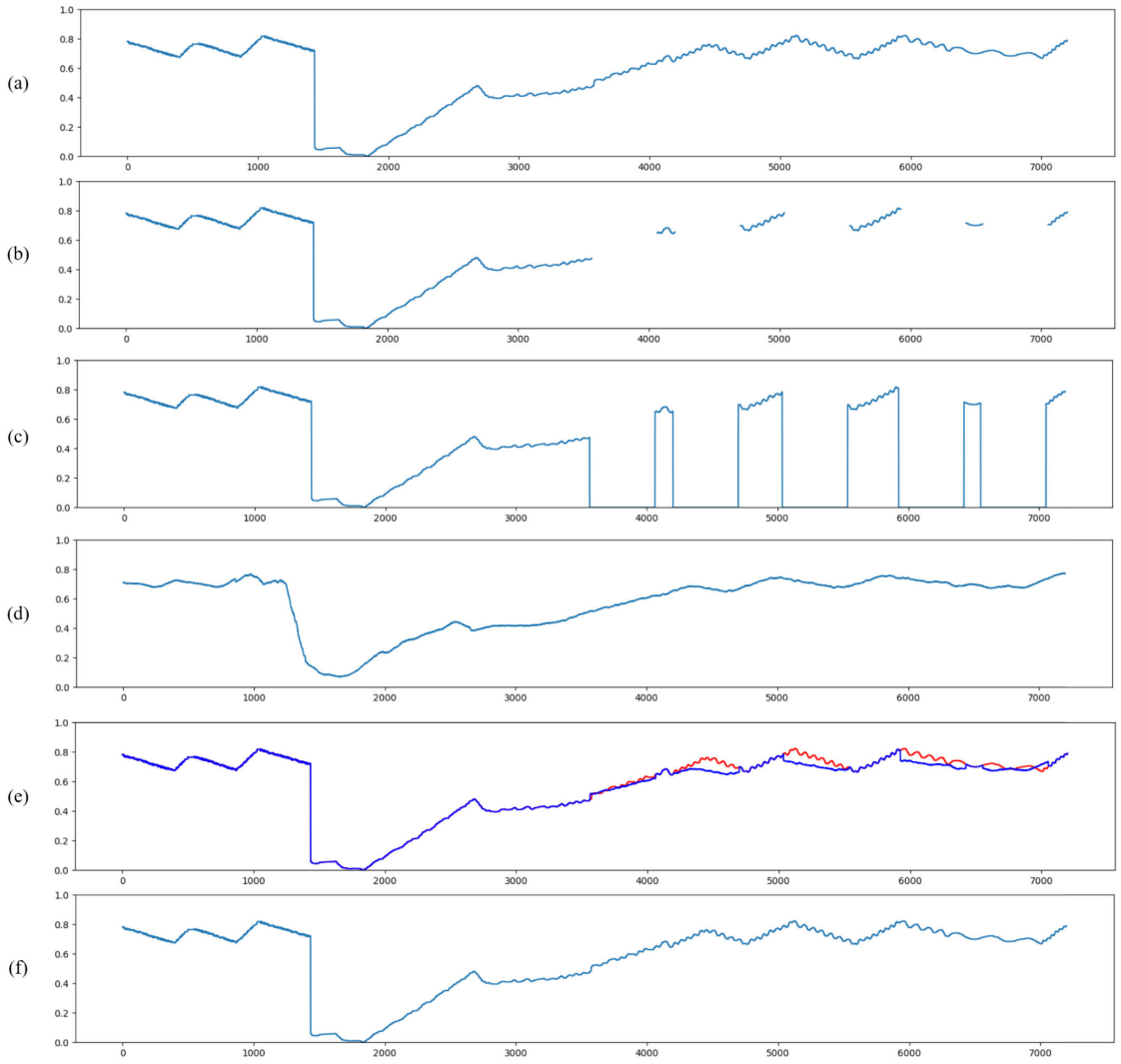

FIGURE 3. The presented method's output is illustrated through a series of visual representations: (a) depicts the original signal, serving as the expected output; (b) displays the signal with randomly generated range missing values; (c) exhibits the signal with zero-value imputation; (d) showcases the reconstructed output achieved through the proposed methodology; (e) demonstrates the incorporation of the reconstructed signal (from (d)) into the missing value positions (from (b)); and finally, (f) presents the ultimate output of the proposed method in blue, contrasted with the expected output in red for comparative analysis.

0.0024 ± 0.0010, as shown in Tables 7 and 8. This outcome is expected, as the Hot-Deck technique fills in missing values based on the most recent available data, leading to near-perfect imputation of signals with Gaussian-style loss, provided their characteristics exhibit limited variance.

Conversely, the Hot-Deck method struggles significantly with extensive data loss, particularly in cases of range missing data. This is due to its tendency to fill in missing values with the most recent data, introducing biases and resulting in irregular signal shapes.

Consequently, our method remains effective, particularly in scenarios with significant data loss, reflecting its applicability across various industries. This effectiveness is supported by scores from RMSE and N-DTW metrics; lower scores indicate better preservation of the shape, position of anomalies, and overall characteristics by the imputation methods used.

These observations are evident in the case of LIT002 with 90% range missing data (Tables 1 and 3), where the Hot-Deck method's RMSE and N-DTW scores are 0.0440 ± 0.0030 and 19.4883 ± 0.7796, respectively, while the proposed method achieves 0.0145 ± 0.0040 and 8.1388 ± 0.8048. Similarly, in the LIT401 experiment under the same conditions (Tables 7 and 8), the Hot-Deck method shows RMSE and N-DTW scores of 0.0229 ± 0.0009 and 14.5425 ± 0.3347, while the proposed method achieves 0.0128 ± 0.0015 and 10.5954 ± 0.7159.

It is crucial to emphasize that in real-world scenarios, downtime or data loss often exhibits a pattern within a

TABLE 3. A comparison of the N-DTW results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	9.5946 ± 0.1501	3.3834 ± 0.1598	4.7842 ± 0.0269	—	—
	Zero	—	—	—	3.1420 ± 0.2702	2.7534 ± 0.3139
	Random	—	—	—	2.4257 ± 0.0937	$\mathbf{2.2366 \pm 0.1378}$
	Mean	—	—	—	2.6753 ± 0.1492	2.4247 ± 0.1822
50 % Missing (range)	NaN	15.5474 ± 0.2642	10.6973 ± 0.3306	9.7199 ± 1.5712	—	—
	Zero	—	—	—	7.6832 ± 0.5961	7.0045 ± 0.6606
	Random	—	—	—	6.1622 ± 0.4796	5.4561 ± 0.4389
	Mean	—	—	—	6.8254 ± 0.3072	$\mathbf{5.2575 \pm 0.5603}$
90 % Missing (range)	NaN	21.6153 ± 0.2755	19.4883 ± 0.7796	11.5343 ± 0.0229	—	—
	Zero	—	—	—	10.8219 ± 0.3906	$\mathbf{8.1688 \pm 0.8048}$
	Random	—	—	—	18.1804 ± 1.5599	9.7894 ± 1.1900
	Mean	—	—	—	12.8495 ± 0.8356	9.9119 ± 1.4965
20 % Missing (Gaussian)	NaN	9.8326 ± 0.1150	$\mathbf{0.2207 \pm 0.0092}$	4.7536 ± 0.0084	—	—
	Zero	—	—	—	12.7888 ± 0.9756	10.3181 ± 1.5754
	Random	—	—	—	2.0297 ± 0.2915	1.8408 ± 0.2803
	Mean	—	—	—	1.9760 ± 0.1296	1.6983 ± 0.2613
50 % Missing (Gaussian)	NaN	15.5620 ± 0.1613	$\mathbf{0.5510 \pm 0.0289}$	7.5438 ± 0.0055	—	—
	Zero	—	—	—	5.4325 ± 0.8609	5.8556 ± 0.7567
	Random	—	—	—	5.8899 ± 0.5934	5.4935 ± 1.1814
	Mean	—	—	—	5.8230 ± 0.8318	5.6094 ± 1.7023
90 % Missing (Gaussian)	NaN	20.8599 ± 0.2424	$\mathbf{1.9244 \pm 0.0720}$	10.2464 ± 0.0039	—	—
	Zero	—	—	—	12.3393 ± 1.0812	10.1511 ± 1.3601
	Random	—	—	—	11.4515 ± 0.9283	9.9419 ± 1.3677
	Mean	—	—	—	12.1028 ± 1.0037	10.0408 ± 2.0013

TABLE 4. A comparison of the PRECISION results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9904 ± 0.0140	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	0.9998 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9998 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.8922 ± 0.0266	0.9954 ± 0.0065	0.9613 ± 0.0096	—	—
	Zero	—	—	—	0.9977 ± 0.0024	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9987 ± 0.0023	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9959 ± 0.0048	0.9929 ± 0.0214
90 % Missing (range)	NaN	0.3000 ± 0.1500	0.9459 ± 0.0165	0.9379 ± 0.0261	—	—
	Zero	—	—	—	0.9999 ± 0.0003	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9997 ± 0.0009	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	0.9981 ± 0.0058	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.8619 ± 0.0790	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.2500 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

range, frequently involving substantial levels of missing data. Therefore, our proposed method is well-suited for addressing challenges in industrial settings susceptible to cyberattacks, where attackers seek to inflict persistent and severe damage to the system.

As to be expected, the results shows that the proposed method gain superiority in most aspect and cases. The error of the imputed signal compares the ground truth using the proposed method are better in both RMSE and N-DTW aspect. As for the accuracy, precision, recall,

TABLE 5. A comparison of the RECALL results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9898 ± 0.0158	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0034}$	—	—
	Zero	—	—	—	0.9998 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9998 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.8597 ± 0.0426	0.9953 ± 0.0067	0.9578 ± 0.0113	—	—
	Zero	—	—	—	0.9976 ± 0.0025	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9987 ± 0.0023	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9959 ± 0.0050	0.9917 ± 0.0249
90 % Missing (range)	NaN	0.5000 ± 0.0001	0.9386 ± 0.0210	0.9271 ± 0.0338	—	—
	Zero	—	—	—	0.9999 ± 0.0003	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9997 ± 0.0009	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	0.9980 ± 0.0060	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.7758 ± 0.1555	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.5000 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

TABLE 6. A comparison of the F1-SCORE results based on the LIT002 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9897 ± 0.0158	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	0.9997 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9999 ± 0.0002	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9997 ± 0.0005	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.8561 ± 0.0452	0.9952 ± 0.0067	0.9578 ± 0.0113	—	—
	Zero	—	—	—	0.9976 ± 0.0025	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9987 ± 0.0023	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	0.9958 ± 0.0050	0.9916 ± 0.0251
90 % Missing (range)	NaN	0.3334 ± 0.0003	0.9383 ± 0.0213	0.9265 ± 0.0344	—	—
	Zero	—	—	—	0.9999 ± 0.0003	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9997 ± 0.0009	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	0.9980 ± 0.0060	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.7416 ± 0.1976	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.3333 ± 0.0000	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

f1-score perspective, the proposed model also outperformed baseline as well. Though, in some 20% and 5 Ranges cases, the performance might be slightly poorer compared to the baseline.

The cornerstone of this research's success stems from the innovative data imputation method, which harnesses the power of Variational Autoencoders (VAE) and latent space coordination. It is noteworthy that while the baseline framework also employs VAE, it primarily fails to capture the intricate interplay between sensor data. Additionally, limited data samples pose a significant challenge for the VAE in the baseline method, hindering its capacity to effectively learn

TABLE 7. A comparison of the RMSE results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.0053 ± 0.0000	0.0014 ± 0.0001	0.0142 ± 0.0093	—	—
	Zero	—	—	—	0.0033 ± 0.0006	0.0026 ± 0.0004
	Random	—	—	—	0.0015 ± 0.0001	0.0012 ± 0.0002
	Mean	—	—	—	0.0010 ± 0.0002	$\mathbf{0.0009 \pm 0.0002}$
50 % Missing (range)	NaN	0.0137 ± 0.0001	0.0091 ± 0.0005	0.0107 ± 0.0001	—	—
	Zero	—	—	—	0.0119 ± 0.0009	0.0081 ± 0.0008
	Random	—	—	—	0.0080 ± 0.0013	0.0057 ± 0.0010
	Mean	—	—	—	0.0059 ± 0.0005	$\mathbf{0.0054 \pm 0.0011}$
90 % Missing (range)	NaN	0.0250 ± 0.0001	0.0229 ± 0.0009	0.0263 ± 0.0001	—	—
	Zero	—	—	—	0.0196 ± 0.0015	0.0143 ± 0.0008
	Random	—	—	—	0.0431 ± 0.0787	$\mathbf{0.0128 \pm 0.0015}$
	Mean	—	—	—	0.0224 ± 0.0050	0.0136 ± 0.0018
20 % Missing (Gaussian)	NaN	0.0052 ± 0.0000	$\mathbf{0.0000 \pm 0.0000}$	0.0370 ± 0.0017	—	—
	Zero	—	—	—	0.0048 ± 0.0133	0.0003 ± 0.0001
	Random	—	—	—	0.0005 ± 0.0003	0.0003 ± 0.0001
	Mean	—	—	—	0.0003 ± 0.0001	0.0003 ± 0.0002
50 % Missing (Gaussian)	NaN	0.0130 ± 0.0001	$\mathbf{0.0000 \pm 0.0000}$	0.0094 ± 0.0001	—	—
	Zero	—	—	—	0.0024 ± 0.0006	0.0020 ± 0.0006
	Random	—	—	—	0.0026 ± 0.0013	0.0024 ± 0.0010
	Mean	—	—	—	0.0034 ± 0.0010	0.0026 ± 0.0008
90 % Missing (Gaussian)	NaN	0.0235 ± 0.0001	$\mathbf{0.0004 \pm 0.0000}$	0.0161 ± 0.0091	—	—
	Zero	—	—	—	0.0181 ± 0.0032	0.0102 ± 0.0032
	Random	—	—	—	0.0187 ± 0.0045	0.0122 ± 0.0030
	Mean	—	—	—	0.0218 ± 0.0063	0.0119 ± 0.0037

TABLE 8. A comparison of the N-DTW results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	7.2665 ± 0.0355	2.6935 ± 0.1082	9.8463 ± 3.8879	—	—
	Zero	—	—	—	4.6419 ± 0.4499	3.8396 ± 0.3175
	Random	—	—	—	2.9167 ± 0.1695	2.4750 ± 0.2934
	Mean	—	—	—	2.3928 ± 0.2249	$\mathbf{2.1933 \pm 0.3536}$
50 % Missing (range)	NaN	11.6911 ± 0.0465	8.3942 ± 0.2738	8.7330 ± 0.0492	—	—
	Zero	—	—	—	9.9412 ± 0.4199	7.9282 ± 0.4224
	Random	—	—	—	8.0442 ± 0.6882	6.3838 ± 0.5507
	Mean	—	—	—	6.8757 ± 0.3612	$\mathbf{6.2953 \pm 0.7014}$
90 % Missing (range)	NaN	15.8007 ± 0.0402	14.5425 ± 0.3347	14.7640 ± 0.0372	—	—
	Zero	—	—	—	13.5004 ± 0.4600	11.0938 ± 0.3846
	Random	—	—	—	16.6101 ± 12.0719	$\mathbf{10.5954 \pm 0.7159}$
	Mean	—	—	—	14.3063 ± 1.3929	10.9698 ± 0.7991
20 % Missing (Gaussian)	NaN	7.2174 ± 0.0133	$\mathbf{0.1946 \pm 0.0070}$	5.2876 ± 0.0146	—	—
	Zero	—	—	—	3.7829 ± 5.7729	1.5984 ± 0.3018
	Random	—	—	—	1.9727 ± 0.5779	1.5370 ± 0.3392
	Mean	—	—	—	1.7126 ± 0.3133	1.6621 ± 0.4954
50 % Missing (Gaussian)	NaN	11.4141 ± 0.0235	$\mathbf{0.5301 \pm 0.0104}$	8.3938 ± 0.0064	—	—
	Zero	—	—	—	4.6511 ± 0.5426	4.0440 ± 0.6913
	Random	—	—	—	4.7660 ± 1.0329	4.4328 ± 0.9065
	Mean	—	—	—	5.4449 ± 0.8128	4.5877 ± 0.8119
90 % Missing (Gaussian)	NaN	15.3221 ± 0.0289	$\mathbf{1.8034 \pm 0.0173}$	9.8059 ± 3.8879	—	—
	Zero	—	—	—	12.9672 ± 1.1071	9.4070 ± 1.4015
	Random	—	—	—	13.0075 ± 1.3993	10.2430 ± 1.2665
	Mean	—	—	—	13.8927 ± 1.6912	10.2114 ± 1.4381

and exploit the latent space during the imputation process. In contrast, our proposed model not only adeptly predicts latent space relationships but also excels in reconstructing the original data, resulting in imputed signals that closely mirror their original counterparts in terms of quality and structural fidelity.

It is imperative to acknowledge that in a limited subset of cases, constituting approximately 20% of instances falling

TABLE 9. A comparison of the ACCURACY results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	99.7650 ± 0.4266	99.9000 ± 0.1140	94.7240 ± 4.4754	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
50 % Missing (range)	NaN	98.2250 ± 1.7131	97.5950 ± 1.0083	92.2250 ± 0.2205	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	99.9950 ± 0.0150
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
90 % Missing (range)	NaN	78.2050 ± 16.8337	89.8100 ± 2.4300	89.8300 ± 0.2159	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	95.0000 ± 15.0000	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	97.2550 ± 7.7570	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	80.7150 ± 23.0213	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{100.0000 \pm 0.0000}$	$\mathbf{100.0000 \pm 0.0000}$

TABLE 10. A comparison of the PRECISION results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9977 ± 0.0042	0.9990 ± 0.0011	0.9553 ± 0.0374	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.9834 ± 0.0151	0.9772 ± 0.0093	0.9327 ± 0.0016	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (range)	NaN	0.8163 ± 0.2006	0.9161 ± 0.0177	0.9155 ± 0.0015	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.00000}$
	Random	—	—	—	0.9250 ± 0.2250	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.9815 ± 0.0509	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.00000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.7995 ± 0.2890	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

within specific value ranges and some range missing data cases, our method's performance may exhibit slight deviations when compared to the baseline. These disparities can be attributed to the nuances inherent in the signal reconstruction process, stemming from the decoder's capabilities and the neural network's latent predictions. These deviations tend to affect finer details of the signal, as the framework primarily learns the prominent data characteristics. Consequently, in certain scenarios, our model may not capture the signal with the desired precision. Hence, it is reasonable to infer that our framework excels in extreme cases where substantial data reconstruction is required.

TABLE 11. A comparison of the RECALL results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9976 ± 0.0043	0.9990 ± 0.0011	0.9472 ± 0.0448	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.9822 ± 0.0171	0.9759 ± 0.0101	0.9223 ± 0.0022	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (range)	NaN	0.7821 ± 0.1683	0.8981 ± 0.0243	0.8983 ± 0.0022	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9500 ± 0.1500	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.9726 ± 0.0776	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.8072 ± 0.2302	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

The industrial sector often faces the imminent risk of severe data loss due to anomalies, which can extend to the disruption of sensors and actuators responsible for data acquisition. As a consequence, the available reported data may contain only a scant number of data samples. It is precisely within these challenging scenarios that our framework reveals its remarkable potential, proficiently restoring the majority, if not the entirety, of the lost data close to its original, unaltered form.

To summarize the advantages and limitations of our proposed method, the Anomaly Signal Imputation Using Latent Coordination Relations is specifically designed for industrial data containing anomalies such as cyberattacks. This framework excels in cases where signal data is substantially and continuously lost, as it effectively utilizes the latent space relations based on the training model. However, in scenarios where signal loss is minimal or follows a Gaussian distribution, simpler methods may perform better. In real-world situations involving cyberattacks, continuous data loss is more common than random noise, highlighting the practical relevance of our framework.

VI. CONCLUSION

In conclusion, the presence of missing data poses significant challenges in data analysis, particularly in the context of anomalous events. Traditional methods and existing techniques have struggled to effectively restore the original characteristics of the data and handle heavily missing data. To overcome these limitations, we have introduced a novel framework named Anomaly Signal Imputation Using Latent Coordination Relations. This framework leverages a variational autoencoder (VAE) to learn from complete data and capture the latent space representation. By extracting coordination points from the latent space, we establish a prediction model for data imputation. Experimental evaluations conducted on anomalous signals from a water treatment testbed demonstrate the superiority of our proposed method, outperforming baseline techniques in most scenario, specifically in the highly loss of signal data cases. Moreover, the proposed framework significantly enhances the similarity of the output signals. This success can be attributed to the restorative capabilities of the VAE's decoder and the framework's ability to uncover relationships among individual signals. Our approach offers a promising solution to address missing data challenges in the presence of anomalous events, contributing to advancements in data imputation techniques.

Future research should explore the broader application of latent coordination relations. While our framework has proven effective in signal data, extending its use to domains such as image and video analysis is essential. Experiments involving diverse anomaly cases beyond signal data will provide valuable insights into the framework's versatility and applicability. Additionally, it is crucial to investigate the impact of various anomaly types and develop a modification model capable of detecting multiple anomalous incidents to enhance real-world applicability. Moreover, fields such as radio astronomy research [50] may benefit from applying the proposed framework, with appropriate modifications, to improve data imputation across various applications.

TABLE 12. A comparison of the F1-SCORE results based on the LIT401 dataset.

Missing Type	Fill Type	Mean Method	Hot Deck Method	KNN Method	VAE Method	Proposed Method
20 % Missing (range)	NaN	0.9976 ± 0.0043	0.9990 ± 0.0011	0.9468 ± 0.0452	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (range)	NaN	0.9822 ± 0.0172	0.9759 ± 0.0101	0.9218 ± 0.0022	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	0.9999 ± 0.0002
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (range)	NaN	0.7414 ± 0.2279	0.8969 ± 0.0249	0.8972 ± 0.0022	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	0.9333 ± 0.2000	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
20 % Missing (Gaussian)	NaN	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
50 % Missing (Gaussian)	NaN	0.9707 ± 0.0832	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
90 % Missing (Gaussian)	NaN	0.7510 ± 0.3009	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$	—	—
	Zero	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Random	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
	Mean	—	—	—	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$

APPENDIX
SUPPLEMENTARY EXPERIMENTS
A. THE EXPERIMENTAL RESULTS BASED ON THE LIT002 DATASET

The experimental results for imputing LIT002 sensor data are shown in Tables 3 through 6.

B. THE EXPERIMENTAL RESULTS BASED ON THE LIT401 DATASET

The experimental results for imputing LIT401 sensor data are shown in Tables 7 through 12.

REFERENCES

[1] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, Mar. 2021, doi: 10.1145/3439950.

[2] G. Li and J. J. Jung, "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges," *Inf. Fusion*, vol. 91, pp. 93–102, Mar. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522001774

[3] K. Takahashi, R. Ooka, and S. Ikeda, "Anomaly detection and missing data imputation in building energy data for automated data pre-processing," in *Proc. J. Phys., Conf.*, Nov. 2021, vol. 2069, no. 1, Art. no. 012144, doi: 10.1088/1742-6596/2069/1/012144.

[4] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *J. Big Data*, vol. 8, no. 1, p. 140, Oct. 2021.

[5] R. J. Mislevy, R. J. A. Little, and D. B. Rubin, "Statistical analysis with missing data," *J. Educ. Stat.*, vol. 16, no. 2, p. 150, 1991.

[6] H. Yu, S. E. Perumean-Chaney, and K. A. Kaiser, "What is missing in missing data handling? An evaluation of missingness in and potential remedies for doctoral dissertations and subsequent publications that use NHANES data," *J. Statist. Data Sci. Educ.*, vol. 32, no. 1, pp. 3–10, Apr. 2023, doi: 10.1080/26939169.2023.2177214.

[7] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of performance of data imputation methods for numeric dataset," *Appl. Artif. Intell.*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.

[8] Q. Song and M. Shepperd, "Missing data imputation techniques," *Int. J. Bus. Intell. Data Mining*, vol. 2, no. 3, p. 261, 2007.

[9] L. O. Joel, W. Doorsamy, and B. S. Paul, "A review of missing data handling techniques for machine learning," *Int. J. Innov. Technol. Interdiscipl. Sci.*, vol. 5 no. 3, pp. 971–1005, 2022. [Online]. Available: https://www.ijitis.org/index.php/ijitis/article/view/94

[10] S. Z. Christopher, T. Siswantining, D. Sarwinda, and A. Bustaman, "Missing value analysis of numerical data using fractional hot deck imputation," in *Proc. 3rd Int. Conf. Informat. Comput. Sci. (ICICoS)*, Oct. 2019, pp. 1–6.

[11] W. Zhao-hong, "Numeric missing value's hot deck imputation based on cloud model and association rules," in *Proc. 2nd Int. Workshop Educ. Technol. Comput. Sci.*, vol. 1, Mar. 2010, pp. 238–241.

[12] K. I. Penny, M. Zegham Ashraf, and J. C. Duffy, "The use of hot deck imputation to compare performance of further education colleges," in *Proc. 29th Int. Conf. Inf. Technol. Interfaces*, Jun. 2007, pp. 179–184.

[13] L. Beretta and A. Santaniello, "Nearest neighbor imputation algorithms: A critical evaluation," *BMC Med. Inform. Decis. Making*, vol. 16, no. S3, p. 74, Jul. 2016.

[14] G. Batista and M.-C. Monard, "A study of K-nearest neighbour as an imputation method," in *Proc. Int. Conf. Health Inf. Sci.*, vol. 30, Jan. 2002, pp. 251–260.

[15] Y. Ang, Y. Qian, and S. Gao, "Factory energy data imputation by nearest neighbor search with clustering," in *Proc. IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. (AEECA)*, Aug. 2020, pp. 302–307.

[16] J. T. McCoy, S. Kroon, and L. Auret, "Variational autoencoders for missing data imputation with application to a simulated milling circuit," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 141–146, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318320949

[17] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2022, *arXiv:1312.6114*.

[18] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2017, pp. 88–99.

[19] (Nov. 12, 2023). *Itrust—Singapore University of Technology and Design (Sutd)*. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/

[20] H. Kang, "The prevention and handling of the missing data," *Korean J. Anesthesiol.*, vol. 64, no. 5, p. 402, 2013.

[21] A. E. Karrar, "The effect of using data pre-processing by imputations in handling missing values," *Indonesian J. Elect. Eng. Inform.*, vol. 10, no. 2, pp. 375–384, Apr. 2022, doi: 10.52549/ijeei.v10i2.3730.

[22] S. Reddy Sankepally, N. Kosaraju, and K. Mallikharjuna Rao, "Data imputation techniques: An empirical study using chronic kidney disease and life expectancy datasets," in *Proc. Int. Conf. Innov. Trends Inf. Technol. (ICITIIT)*, Feb. 2022, pp. 1–7.

[23] J. L. Schafer and J. W. Graham, "Missing data: Our view of the state of the art," *Psychol. Methods*, vol. 7, no. 2, pp. 147–177, 2002.

[24] P. Ranganathan and S. Hunsberger, "Handling missing data in research," *Perspect. Clin. Res.*, vol. 15, no. 2, pp. 99–101, Apr. 2024.

[25] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, May 2019, pp. 1255–1260.

[26] S. Thirukumaran and A. Sumathi, "Improving accuracy rate of imputation of missing data using classifier methods," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–7.

[27] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, nos. 5–6, pp. 519–533, May 2003, doi: 10.1080/713827181.

[28] M. Saad, M. Chaudhary, L. Nassar, F. Karray, and V. Gaudet, "Versatile deep learning based application for time series imputation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[29] J. Kwon, C. Cha, and H. Park, "Multilayered LSTM with parameter transfer for vehicle speed data imputation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[30] Z. Zhou, J. Mo, and Y. Shi, "Data imputation and dimensionality reduction using deep learning in industrial data," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 2329–2333.

[31] Y. Duan, Y. Lv, W. Kang, and Y. Zhao, "A deep learning based approach for traffic data imputation," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 912–917.

[32] S. Phung, A. Kumar, and J. Kim, "A deep learning technique for imputing missing healthcare data," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 6513–6516.

[33] E. C. Erkus, M. Ersel Er, A. Yildiz, and M. Gencer, "An investigation of the effects of the numerical missing value imputation methods for click-through rate estimation performances," in *Proc. 11th Int. Symp. Digit. Forensics Secur. (ISDFS)*, May 2023, pp. 1–5.

[34] D. E. Diamantis, P. Gatoula, and D. K. Iakovidis, "EndoVAE: Generating endoscopic images with a variational autoencoder," in *Proc. IEEE 14th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jun. 2022, pp. 1–5.

[35] L. Cai, H. Gao, and S. Ji, "Multi-stage variational auto-encoders for coarse-to-fine image generation," in *Proc. SIAM Int. Conf. Data Mining*, May 2019, pp. 630–638.

[36] A. Islam and S. B. Belhaouari, "Fast and efficient image generation using variational autoencoders and K-Nearest neighbor OveRsampling approach," *IEEE Access*, vol. 11, pp. 28416–28426, 2023.

[37] Z.-S. Liu, W.-C. Siu, and Y.-L. Chan, "Photo-realistic image super-resolution via variational autoencoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1351–1365, Apr. 2021.

[38] D. Chira, I. Haralampiev, O. Winther, A. Dittadi, and V. Liévin, "Image super-resolution with deep variational autoencoders," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2023, pp. 395–411.

[39] J. Xu and Y. Zhao, "Image super-resolution based on variational autoencoder and channel attention," in *Proc. 6th Int. Conf. Artif. Intell. Pattern Recognit.*, New York, NY, USA, 2024, pp. 611–616, doi: 10.1145/3641584.3641675.

[40] R. R. Andridge and R. J. A. Little, "A review of hot deck imputation for survey non-response," *Int. Stat. Rev.*, vol. 78, no. 1, pp. 40–64, Apr. 2010.

[41] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.

[42] A. Hemmati-Sarapardeh, A. Larestani, M. Nait Amar, and S. Hajirezaie, *Introduction*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 1–22, doi: 10.1016/B978-0-12-818680-0.00001-1.

[43] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

[44] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

[45] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022, doi: 10.1038/s41598-022-09954-8.

[46] S. Akter, A. Habib, Md. A. Islam, Md. S. Hossen, W. A. Fahim, P. R. Sarkar, and M. Ahmed, "Comprehensive performance assessment of deep learning models in early prediction and risk identification of chronic kidney disease," *IEEE Access*, vol. 9, pp. 165184–165206, 2021, doi: 10.1109/ACCESS.2021.3129491.

[47] A. Y. Yildiz, E. Koç, and A. Koç, "Multivariate time series imputation with transformers," *IEEE Signal Process. Lett.*, vol. 29, pp. 2517–2521, 2022.

[48] I. Azimi, T. Pahikkala, A. M. Rahmani, H. Niela-Vilén, A. Axelin, and P. Liljeberg, "Missing data resilient decision-making for healthcare IoT through personalization: A case study on maternal health," *Future Gener. Comput. Syst.*, vol. 96, pp. 297–308, Jul. 2019.

[49] R. Wu, S. D. Hamshaw, L. Yang, D. W. Kincaid, R. Etheridge, and A. Ghasemkhani, "Data imputation for multivariate time series sensor data with large gaps of missing data," *IEEE Sensors J.*, vol. 22, no. 11, pp. 10671–10683, Jun. 2022.

[50] E. Ghaderpour, "Least-squares wavelet and cross-wavelet analyses of VLBI baseline length and temperature time series: Fortaleza–Hartebeesthoek–Westford–Wettzell," *Publications Astronomical Soc. Pacific*, vol. 133, no. 1019, Dec. 2020, Art. no. 014502, doi: 10.1088/1538-3873/abcc4e.

THASORN CHALONGVORACHAI received the B.S. and M.S. degrees in information technology from the King Mongkut's Institute of Technology Ladkrabang, in 2018 and 2021, respectively. From 2017 to 2023, he was a Research Assistant with the Image Processing and Deep Learning Laboratory. His research interests include anomaly detection on the signal data, variational autoencoders, and deep learning. His awards and honors include the IEEE 15th International Conference on Information Technology and Electrical Engineering Best Paper Award, in 2023, and a scholarship from the Sakura Science Exchange Program.

KUNTPONG WORARATPANYA (Member, IEEE) received the B.Ind.Tech. degree in computer technology, the M.Eng. degree in computer engineering, and the D.Eng. degree in electrical engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1992, 1996, and 2005, respectively. He is currently an Associate Professor with the School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, and has also chaired the IEEE Computational Intelligence Society Thailand Chapter (2024–2025). His research interests include stereoscopic acquisition and compression, multimedia coding and processing, signal processing, speech recognition and processing, pattern recognition and image processing, computer vision, and machine learning/deep learning.