Investigating a Model-Agnostic and Imputation-Free Approach for Irregularly-Sampled Multivariate Time-Series Modeling

Anonymous authors
Paper under double-blind review

Abstract

Modeling Irregularly-sampled and Multivariate Time Series (IMTS) is crucial across a variety of applications where different sets of variates may be missing at different time-steps due to sensor malfunctions or high data acquisition costs. Existing approaches for IMTS either consider a two-stage impute-then-model framework or involve specialized architectures specific to a particular model and task. We perform a series of experiments to derive insights about the performance of IMTS methods on a variety of semi-synthetic and real-world datasets for both classification and forecasting. We also introduce **Miss**ing Feature-aware **T**ime **S**eries **M**odeling (MissTSM) or MissTSM, a simple model-agnostic and imputation-free approach for IMTS modeling. We show that MissTSM shows competitive performance compared to other IMTS approaches, especially when the amount of missing values is large and the data lacks simplistic periodic structures—conditions common to real-world IMTS applications.

1 Introduction

Deep Learning for modeling multivariate Time-Series (MTS) is a rapidly growing field, with two major downstream tasks: forecasting and classification. Research (Dong et al., 2024; Nie et al., 2022a; Liu et al., 2023) in this field has been fueled by the availability of benchmark MTS datasets spanning diverse applications such as electric load forecasting and health monitoring containing fixed sets of variates regularly sampled over time. However, real-world MTS applications are plagued by missing values occurring over arbitrary sets of variates at every time-step (e.g. due to sensor malfunctions), resulting in Irregularly-sampled MTS (IMTS) datasets. IMTS modeling is particularly challenging because the misalignment of variates across time impairs transformer models that assume a fixed set of variates to be observed at every time-step

A common approach for IMTS modeling is to use a two-step framework where we first use imputation methods (Ahn et al., 2022; Batista et al., 2002) to fill in missing values based on observed data, followed by feeding the imputed time-series to an MTS model (see Figure 1). Note that the choice of imputation method is agnostic to the MTS model, making it "model-agnostic." However, the effectiveness of this framework relies on the quality of performed imputation, which can degrade if the time-series lacks periodic structure or if the imputation method is overly simplistic. Imputation can also introduce artificial patterns or artifacts into the data, which MTS models may interpret as genuine trends or observations. Moreover, deep learning-based imputation methods require training, which adds to the overall computational cost of IMTS modeling.

Imputation-free approaches for IMTS have also been developed in recent literature (Che et al., 2018; Rubanova et al., 2019), that involve specialized architectures to handle missing values in time-series for specific downstream tasks such as classification (see Figure 1) However, these approaches have been empirically shown to struggle with capturing long-term temporal dependencies that are central to the problem of forecasting, are difficult to parallelize, and incur high computational costs. Furthermore, existing imputation-free IMTS approaches are not model-agnostic, i.e., they have been developed as specialized model architectures that cannot be used as a generic wrapper with latest advances in MTS models, restricting their adaptability and performance.

Given the complementary strengths and weaknesses of existing approaches in IMTS modeling, we ask the following question - *can we develop a model-agnostic and imputation-free approach for IMTS modeling that can be used in a variety of downstream tasks (e.g., forecasting and classification)*? To address this question, we analyze the prevailing strategies for converting time-series into tokens in existing transformer-based MTS models. There are two primary ways for converting time series into tokens: (a) treating each time step (all variates) as a token, or (b) considering each variate (all time steps) as a token (Liu et al., 2023). While these strategies work well for regular MTS data, they are not suited to handle IMTS data because we may be missing some variates at a time step or some time-steps for a variate, making the computation of tokens infeasible. In contrast, we explore a different perspective for creating time-series tokens: independently embedding each combination of time-step and variate as a token. Time-variate combinations with missing values can then be handled using masked cross-attention without performing any explicit imputation. Building on this intuition, we introduce MissTSM, a simple model-agnostic and imputation-free approach for IMTS modeling, designed as a "plug-and-play" layer that can be integrated into any backbone MTS model to handle IMTS data. The advantage of such an approach is that it *(a) does not introduce any imputation artifacts*, and *(b) can act as a wrapper around any MTS model.*

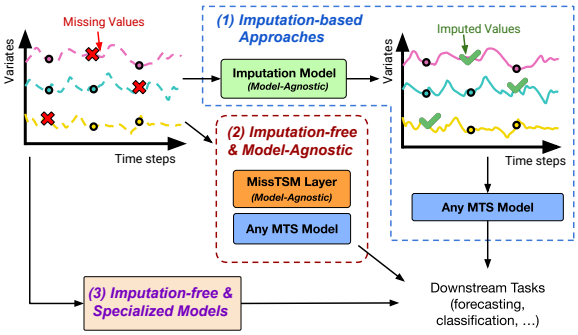


Figure 1: We investigate the relative importance of three categories of approaches for modeling irregular and multivariate time-series: (1) imputation-based approaches, (2) model-agnostic and imputation-free approaches (proposed MissTSM layer), and (3) imputation-free approaches involving specialized architectures.

In this work, we make the following contributions: **(1)** We introduce MissTSM, a model-agnostic and imputation-free approach; **(2)** We conduct a comprehensive experimental study on a variety of datasets for both classification and forecasting tasks, using synthetic masking techniques as well as real-world occurrence of missing values. This study investigates *(a) sensitivity of imputation-based frameworks on the choice of imputation technique and the nature of missing values*, and *(b) the performance of IMTS approaches as the fraction of missing values varies*; **(3)** We demonstrate that MissTSM achieves competitive performance compared to other IMTS approaches, especially when the amount of missing values is large and the data lacks simplistic periodic structures - conditions common to real-world IMTS applications.

2 Related Works

Time-series Forecasting. With the introduction of attention mechanisms via transformer models (Vaswani et al., 2017), a number of transformer-based time-series models have been developed in the last few years (Wu et al., 2021; Nie et al., 2022b; Dong et al., 2024; Liu et al., 2023). While Transformer-based models have shown great promise, recently there has been a strong interest in exploring the use of simple linear models for time-series forecasting as well (Zeng et al., 2023; Ekambaram et al., 2023). In addition, with the rise of self-supervised learning-based models such as masked auto-encoders (MAEs) (He et al., 2022), a new category of MAE-style time-series models have emerged (Dong et al., 2024) that have received a lot of recent interest owing to their ability of learning both low-level and high-level representations for varied downstream tasks such as forecasting and classification. However, while these methods can deal with missing values in the temporal domain, they are unable to handle missing values across both variates and time.

Imputation Methods. Traditionally, most imputation techniques for handling missing values in time-series have been based on statistical approaches (Fung, 2006; Batista et al., 2002; Dempster et al., 1977; Mnih & Salakhutdinov, 2007). In recent years, there is a growing trend to use deep learning methods for time-series imputation, such as SAITS (Du et al., 2023), CSDI (Tashiro et al., 2021a), GAIN (Yoon et al., 2018a), and BRITS (Cao et al., 2018). Imputation techniques can be broadly classified into two classes: those that leverage cross-channel correlations (Batista et al., 2002; Acuna & Rodriguez, 2004) and those that exploit temporal dynamics (Box et al., 2015). Recently, deep learning-based approaches for imputation have been developed (Tashiro et al., 2021b; Cini et al., 2021; Liu et al., 2019; Cao et al., 2018; Du et al., 2023), which

can jointly learn the temporal dynamics with cross-channel correlations. These methods, however, rely on a single entangled representation (or hidden state) to model nonlinear dynamics (Woo et al., 2022) which can be a limitation in capturing the multifaceted nature of time-series. Matrix factorization based techniques (Liu et al., 2022) have also been proposed that offer disentangled temporal representations, enhancing the ability to differentiate and model distinct temporal features. While these deep learning-based models are highly efficient during inference, they require additional training time, which add to the already large time complexity of MTS models.

Imputation-free IMTS Models. In the last decade, there has been a significant growth of models and architectures for learning from IMTS data. Some of the simpler approaches to deal with IMTS data involve working with fixed temporal discretization (Marlin et al., 2012; Lipton et al., 2016). The primary drawback with these approaches is that they make ad-hoc choices in terms of discretization window width and aggregation functions within the windows (Shukla & Marlin, 2020). A popular set of approaches for handling IMTS data are recurrence-based approaches, which includes RNN-based methods such as GRU-D (Che et al., 2018). However, GRU-D has limited scalability to long sequences. Other recurrence-based approaches based on Ordinary Differential Equations (ODE) (Chen et al., 2018; Rubanova et al., 2019) provide an effective solution in modeling the continuous time semantics. These methods are however significantly slow and memory-intensive, as they constantly need to apply the ODE solver and solving ODEs require numerical integration, thus making it impractical for long-term forecasting and large datasets.

Transformer-based methods such as ContiFormer (Chen et al., 2023) and mTAN (Shukla & Marlin, 2021) addresses these limitations by explicitly integrating the modeling abilities of Neural ODEs into the attention mechanism and introducing continuous time attention mechanism that learns time embeddings dynamically, respectively. However, despite ContiFormer being a principled and effective approach, solving an ODE for each key and value incurs a high computational cost. Also, while the runtime speed of mTAN is relatively faster, it is however, inherently optimized toward interpolating missing values by learning representations at fixed set of reference points, thus limiting it's extrapolation or forecasting ability.

This is another limitation of IMTS approaches—their evaluation is mostly limited to a single task, most often to time series classification, thus limiting their applicability. ContiFormer performs evaluation on forecasting tasks, however, they consider regular and clean benchmark time-series datasets in their evaluation. Another limitation in terms of evaluation is that the prior works primarily focus on other IMTS models for comparison, completely ignoring the two-stage imputation approach, which is a more common and practical way of dealing with missing-value data. Our work aims to solve these issues by providing a comprehensive comparison against both existing imputation-free and two-stage imputation-based approaches, and proposing a model-agnostic transformation-allowing any task-specific SOTA model to be applied on any irregularly sampled time-series data with minimal data transformations.

3 Proposed Missing Feature Time-Series Modeling (MissTSM) Framework

3.1 Notations and Problem Formulations

Let us represent a multivariate time-series as $\mathbf{X} \in \mathbb{R}^{T \times N}$, where T is the number of time-steps, and N is the dimensionality (number of variates) of the time-series. We assume a subset of variates (or features) to be missing at some time-steps of $\mathbf{X}$, represented in the form of a missing-value mask $\mathcal{M} \in [0,1]^{T \times N}$, where $\mathcal{M}_{(t,d)}$ represents the value of the mask at t-th time-step and d-th dimension. $\mathcal{M}_{(t,d)} = 1$ denotes that the corresponding value in $\mathbf{X}_{(t,d)}$ is missing, while $\mathcal{M}_{(t,d)} = 0$ denotes that $\mathbf{X}_{(t,d)}$ is observed. Furthermore, let us denote $\mathbf{X}_{(t,:)} \in \mathbb{R}^N$ as the multiple variates of the time-series at a particular time-step t, and $\mathbf{X}_{(:,d)} \in \mathbb{R}^T$ as the uni-variate time-series for the variate d. In this paper, we consider two downstream tasks for time-series modeling: forecasting and classification. For forecasting, the goal is to predict the future S time-steps of $\mathbf{X}$ represented as $\mathbf{Y} \in \mathbb{R}^{S \times N}$. Alternatively, for time-series classification, the goal is to predict output labels $\mathbf{Y} \in \{1, 2, ..., C\}$ given $\mathbf{X}$, where C is the number of classes.

3.2 Learning Embeddings for Time-Series with Missing Features

Limitations of Existing Transformer Methods: The first step in time-series modeling using transformer-based architectures is to learn an embedding of the time-series $\mathbf{X}$ that can be sent to the transformer encoder. Traditionally, this is done using an Embedding layer (typically implemented using a multi-layered perceptron) as $\texttt{Embedding} : \mathbb{R}^N \mapsto \mathbb{R}^D$ that maps $\mathbf{X} \in \mathbb{R}^{T \times N}$ to the embedding $\mathbf{H} \in \mathbb{R}^{T \times D}$, where D is the embedding dimension. The Embedding layer operates on every time-step independently such that the set of variates observed at time-step t, $\mathbf{X}_{(t,:)}$, is considered as a single token and mapped to the embedding vector $\mathbf{h}_t \in \mathbb{R}^D$ as $\mathbf{h}_t = \texttt{Embedding}(\mathbf{X}_{(t,:)})$ (see Figure 2(a)). An alternate embedding scheme was recently introduced in the framework of inverted Transformer (iTransformer) (Liu et al., 2023), where the uni-variate time-series for the d-th variate, $\mathbf{X}_{(:,d)}$, is considered as a single token and mapped to the embedding vector: $\mathbf{h}_d = \texttt{Embedding}(\mathbf{X}_{(:,d)})$ (see Figure 2(b)).

While both these embedding schemes have their unique advantages, they are unfit to handle time-series with arbitrary sets of missing values at every time-step. In particular, the input tokens to the Embedding layer of Transformer or iTransformer requires all components of $\mathbf{X}_{(t,:)}$ or $\mathbf{X}_{(:,d)}$ to be observed, respectively. If any of the components in these tokens are missing, we will not be able to compute their embeddings and thus will have to discard either the time-step or the variate, leading to loss of information.

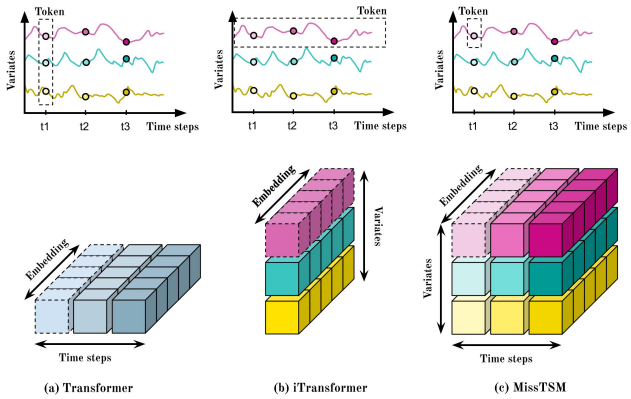


Figure 2: Schematic of the Time-Feature Independent (TFI) Embedding of MissTSM that learns a different embedding for every combination of time-step and variate, in contrast to the time-only embeddings of Transformer (Vaswani et al., 2017) and the variate-only embeddings of iTransformers (Liu et al., 2023).

Time-Feature Independent (TFI) Embedding: To address this challenge as well as to utilize inter-variate interactions similar to Wei et al. (2023), we consider a *Time-Feature Independent (TFI) Embedding* scheme for time-series with missing features, where the value at each combination of time-step t and variate d is considered as a single token $\mathbf{X}_{(t,d)}$, and is independently mapped to an embedding using $\texttt{TFIEmbedding} : \mathbb{R} \mapsto \mathbb{R}^D$ as follows:

$$\mathbf{h}_{(t,d)} = \texttt{TFIEmbedding}(\mathbf{X}_{(t,d)}) \tag{1}$$

In other words, the TFIEmbedding Layer (which is a simple MLP layer) maps $\mathbf{X} \in \mathbb{R}^{T \times N}$ into the TFI embedding $\mathbf{H}^{\text{TFI}} \in \mathbb{R}^{T \times N \times D}$ (see Figure 2(c)). The TFIEmbedding is applied only on tokens $\mathbf{X}_{(t,d)}$ that are observed (for missing tokens, i.e., $\mathcal{M}_{(t,d)} = 1$, we generate a dummy embedding that gets masked out in the MFAA layer). The advantage of such an approach is that even if a particular value in the time-series is missing, other observed values in the time-series can be embedded *"independently"* without being affected by the missing values. Moreover, it allows the MFAA layer to leverage the high-dimensional embeddings to store richer representations bringing in the context of time and variate by computing masked cross-attention among the observed features at a time-step to account for the missing features.

2D Positional Encodings: We add Positional Encoding vectors $\mathbf{PE}$ to the TFI embedding $\mathbf{H}^{\text{TFI}}$ to obtain positionally-encoded embeddings, $\mathbf{Z} = \mathbf{PE} + \mathbf{H}^{\text{TFI}}$. Since TFI embeddings treat every time-feature combination as a token, we use a 2D-positional encoding scheme defined as follows:

$$\begin{aligned} \texttt{PE}(t, d, 2i) &= \sin\left(\frac{t}{10000^{(4i/D)}}\right), \\ \texttt{PE}(t, d, 2i+1) &= \cos\left(\frac{t}{10000^{(4i/D)}}\right) \end{aligned} \tag{2}$$

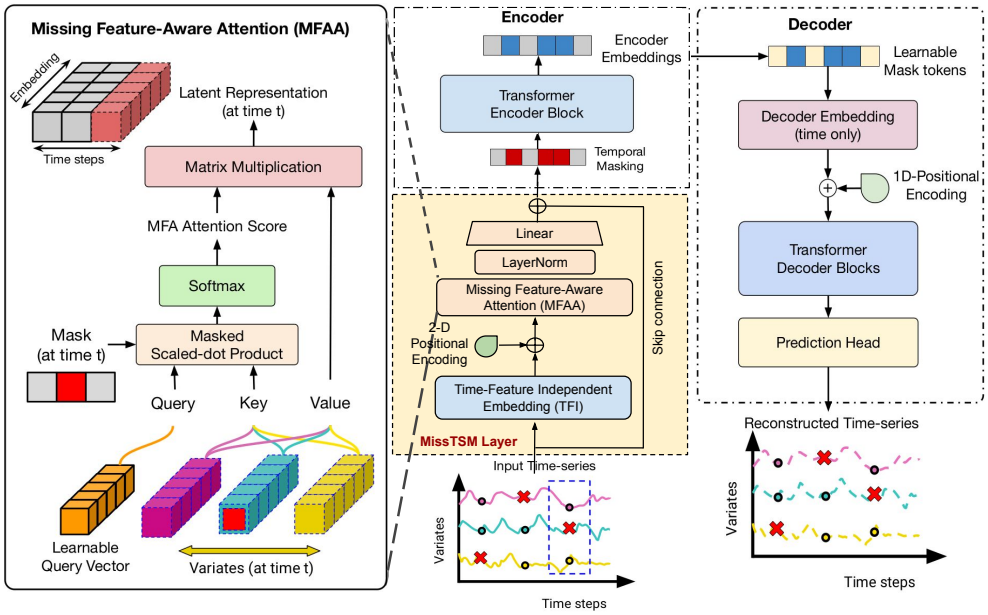


Figure 3: Overview of the MissTSM layer integrated within the Masked Auto-Encoder framework (Li et al., 2023). A zoomed-in view of the MFAA is shown on the left.

$$
\begin{aligned}
\texttt{PE}(t, d, 2j + D/2) &= \sin\left(\frac{d}{10000^{(4j/D)}}\right), \\
\texttt{PE}(t, d, 2j + 1 + D/2) &= \cos\left(\frac{d}{10000^{(4j/D)}}\right)
\end{aligned}
\tag{3}
$$

where t is the time-step, d is the feature, and $i, j \in [0, D/4)$ are integers.

3.3 Missing Feature-Aware Attention (MFAA)

The MFAA Layer illustrated in Figure 3 leverage the power of *"masked-attention"* for learning latent representations at every time-step using partially observed features. MFAA works by computing *attention scores* based on the partially observed features at a time-step t, which are then used to perform a weighted sum of observed features to obtain the latent representation $\mathbf{L}_t$. As shown in Figure 3, these latent representations are projected back using a linear layer, to the original input shape before being fed into the downstream model (here, the encoder-decoder based self-supervised learning framework). Note that MFAA is not designed for long-term temporal modeling. Its primary role is to handle missing values at a time-step by using the observed variates at the same time-step as context. It relies on the subsequent backbone to model long-range temporal dynamics MFAA performs a masked cross-attention using a learnable query vector and observed data as keys and values. This separation of roles is inspired by similar architectures in multi-modal grounding, for example, in Carion et al. (2020), where learnable object queries serve as abstract object representations to focus on distinct objects in an image without requiring predefined region proposals, enabling set-based prediction. Similarly, in our setting, the learnable queries capture the interactions among variates independent of time, enabling the model to attend to the most informative aspects of observed variates at any time-step fed through keys and values. This intuition aligns with the query-based mechanism in mTAN (Shukla & Marlin, 2021), which introduces a structured way to aggregate information over observed time-series data. However, there's is a key difference in the nature of the query - while mTAN uses discrete reference points on a fixed temporal grid to achieve this, our single learnable query generalizes across variates at every time step, allowing for a more flexible representation of feature interactions. Another key difference is in the architecture flow - mTAN first performs a "temporal-first, variate-independent interpolation" to perform temporal attention for each feature to create a regular sequence, then performs a linear combination to

generate a single per timestep representation. In contrast, MissTSM is "variate-first", with the MFAA layer performing cross-variate attention to create a representation L_t at each timestep. Then, the backbone model learns the temporal relationships from this sequence of summaries.

Mathematical Formulation: To obtain attention scores from partially observed features at a time-step, we apply a masked scaled-dot product operation followed by a softmax operation described as follows. We first define a learnable query vector $\mathbf{Q} \in \mathbb{R}^{1\times D}$ which is independent of the variates and time-steps. The positionally-encoded embeddings at time-step t, $\mathbf{Z}_{(t,:)}$, are used as key and value inputs in the MFAA Layer. Specifically, The query, key, and value vectors are defined using linear projections as follows: $\hat{\mathbf{Q}} = \mathbf{Q}\mathbf{W}^{\mathbf{Q}}, \quad \hat{\mathbf{K}}_t = \mathbf{Z}_{(t,:)}\mathbf{W}^{\mathbf{K}}, \quad \hat{\mathbf{V}}_t = \mathbf{Z}_{(t,:)}\mathbf{W}^{\mathbf{V}}$. Here, $\hat{\mathbf{Q}} \in \mathbb{R}^{1\times d_k}$ and $\hat{\mathbf{K}}_t, \hat{\mathbf{V}}_t \in \mathbb{R}^{N\times d_k}$, where d_k is the dimension of the vectors after linear projection. The linear projection matrices for the query, key, and values are defined as: $\mathbf{W}^{\mathbf{Q}}, \mathbf{W}^{\mathbf{K}}, \mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{D\times d_k}$ respectively. Note that the key $\hat{\mathbf{K}}_t$ and value $\hat{\mathbf{V}}_t$ vectors depend on the time-step t, while the query vector doesn't change with time. We then define the Missing Feature-Aware Attention Score at a given time-step t as a masked scalar dot-product of the query and key vector followed by normalization of the scores using a Softmax operation, formally defined as follows:

$$\begin{aligned}\mathbf{A}_t &= \texttt{MFAAScore}(\hat{\mathbf{Q}}, \hat{\mathbf{K}}_t, \mathcal{M}_{(t,:)}) \\ &= \texttt{Softmax}\left(\frac{\hat{\mathbf{Q}}\hat{\mathbf{K}}_t^\top}{\sqrt{d_k}} + \eta\,\mathcal{M}_{(t,:)}\right)\end{aligned} \tag{4}$$

where $\mathbf{A}_t \in \mathbb{R}^N$ is the MFAA Score vector of size N corresponding to the N variates, and $\eta \to -\infty$ is a large negative bias. The negative bias term η forces the masked-elements that correspond to the missing variates in the time-series to have an attention score of zero. Thus, by definition, the i-th element of the MFAA Score $\mathbf{A}_{(t,i)} \neq 0 \implies \mathcal{M}_{(t,:)} = 0$. We compute the latent representation $\mathbf{L}_t$ as a weighted sum of the MFAA score $\mathbf{A}_t$ and the Value vector $\hat{\mathbf{V}}_t$ as follows:

$$\mathbf{L}_t = \texttt{MFAA}(\mathbf{A}_t, \hat{\mathbf{V}}_t) = \mathbf{A}_t\hat{\mathbf{V}}_t \in \mathbb{R}^{d_k} \tag{5}$$

Similar to multi-head attention used in traditional transformers, we extend MFAA to multiple heads as follows:

$$\begin{aligned}&\texttt{MultiHeadMFAA}(\mathbf{Q}, \mathbf{Z}_{(t,:)}, \mathcal{M}_{(t,:)}) \\ &= \text{Concat}(\mathbf{L}_t^0, \mathbf{L}_t^1, \ldots, \mathbf{L}_t^{h-1}) \cdot \mathbf{W}^O\end{aligned} \tag{6}$$

where h is the number of heads, $\mathbf{W^0} \in \mathbb{R}^{hd_k\times D_o}$, $\mathbf{L}_t^i$ is the latent representation obtained from the i-th `MHAA` Layer, and D_o is the output-dimension of the `MultiHeadMFAA` Layer.

3.4 Putting Everything Together: Plugging MissTSM with any MTS Model

Figure 3 shows the overall framework of a Masked Auto-Encoder (MAE) (He et al., 2022) based time-series model integrated with MissTSM. For an input time-series $\mathbf{X}$, we apply the TFI embedding layer followed by the MFAA layer to learn a latent representation for every time-step. The latent representations are then projected back to the original input shape to be fed into the downstream model. In this work, we opted for a MAE-based time-series model as the default downstream or base model, primarily due to its recent success in time-series modeling and its ability to perform both time-series forecasting and classification tasks. Furthermore, out of the several state-of-the-art masked time-series modeling techniques, we intentionally chose the simplest variation of MAE, namely Ti-MAE (Li et al., 2023), to highlight the effectiveness of TFI and MFAA layers in handling missing values.

4 Experimental Setup

Baselines: We benchmark against two categories of models. For MTS, we consider SimMTM (Dong et al., 2024), PatchTST (Nie et al., 2022b), AutoFormer (Wu et al., 2021), DLinear (Zeng et al., 2023), and iTransformer (Liu et al., 2023). Imputation strategies used are, 2nd-order spline interpolation (McKinley &

Levine, 1998), k-Nearest Neighbor (Tan et al., 2019), and SAITS (Du et al., 2023) and BRITS (Cao et al., 2018). For IMTS, we evaluate GRU-D (Che et al., 2018), Latent ODE (Rubanova et al., 2019), SeFT (Horn et al., 2020), mTAND (Shukla & Marlin, 2021), Raindrop (Zhang et al., 2021), and MTGNN (Wu et al., 2020). Baseline choice is aligned with the task each model was originally designed for.

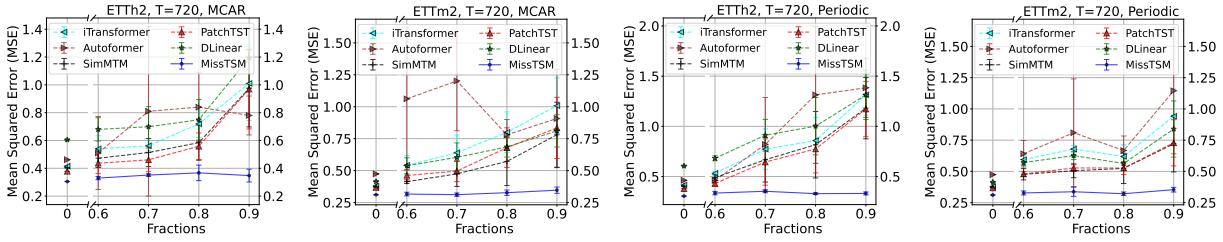
Datasets: We considered three popular time-series forecasting datasets: ETTh2, ETTm2 (Zhou et al., 2021) and Weather (Weather, 2021). For classification, we considered three real-world datasets, namely, Epilepsy (Andrzejak et al., 2001), EMG (Goldberger et al., 2000a), and Gesture (Liu et al., 2009). We follow the same evaluation setups as proposed in TF-C (Zhang et al., 2022). To simulate varying scenarios of missing values appearing in real-world time-series datasets, we adopt two synthetic masking schemes that we apply on these benchmark datasets, namely missing completely at random (MCAR) masking and periodic masking. Furthermore, we compared our performance on five real-world datasets: PhysioNet-2012 (Silva et al., 2012), P12 (Goldberger et al., 2000b) and P19 (Reyna et al., 2020) for health monitoring; Falling Creek Reservoir (FCR) dataset for modeling lake water quality, and Lake Mendota from the North Temperate Lakes Long-Term Ecological Research program (NTL-LTER; Magnuson et al., 2024) also for modeling lakes. See Appendix for more details.

5 Results and Discussions

Here, we discuss our findings with respect to imputation-based vs. imputation free methods, and model-agnostic vs. specialized methods across a variety of datasets, tasks, and missing value settings.

5.1 Imputation-based vs. Imputation-free

5.1.1 Impact of Missing Data Fractions.



(a) ETTh2, MCAR (b) ETTm2, MCAR (c) ETTh2, Periodic (d) ETTm2, Periodic

Figure 4: Performance comparison against different TS Baselines imputed with SAITS, across different missing data fractions.

To understand the effect of varying masking fractions on the forecasting performance, we consider five forecasting models trained on SAITS-imputed data as the set of imputation-based baselines. We compare their results with MissTSM integrated within the MAE framework as an imputation-free approach. Figure 4 shows variations in the Mean Squared Error (MSE) as we increase the missing value fraction in MCAR and periodic masking scheme from 0.6 to 0.9 for forecasting horizon $T = 720$ on two ETT datasets. We can see that, on average, as we increase the amount of missing values in the data, imputation-based baselines and MissTSM show an increasing trend in MSE. This is expected as larger missing value fractions starve IMTS models with greater amount of information degrading their performance. However, the rise in MSE of MissTSM with missing value fractions is much less pronounced than imputation-based baselines consistently across the two datasets and synthetic masking schemes (MCAR and Periodic Masking). Further, note that MissTSM shows smaller standard deviations compared to the large and varying standard deviations of the imputation-based approaches (w.r.t the increasing missingness). These results suggest that imputation-based frameworks struggle when the amount of missing values is high, possibly due to the poor performance of imputation methods when the number of observations is small.

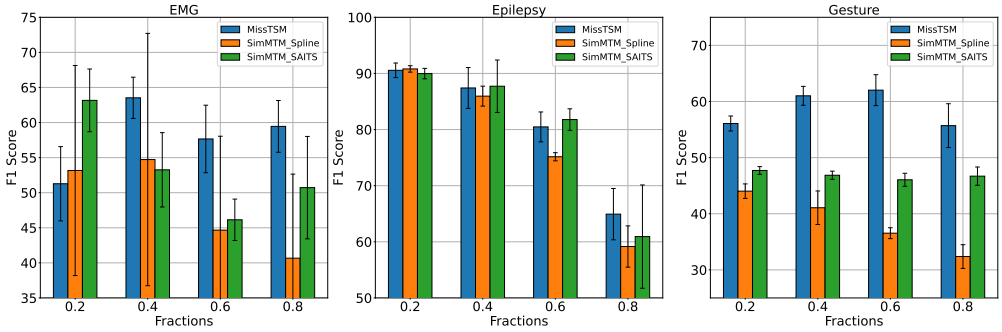


Figure 5: Classification F1 scores on three datasets, EMG, Epilepsy, and Gesture. Masking fractions considered: 0.2, 0.4, 0.6, 0.8.

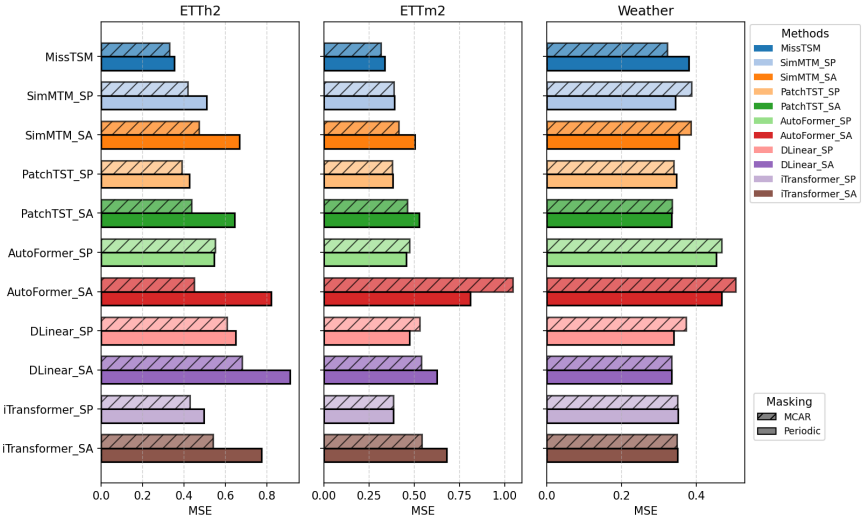


Figure 6: Comparison of different masking methods (70% missing fraction): MCAR vs. Periodic Masking for ETTh2, ETTm2, and Weather datasets. SA stands for SAITS and SP stands for Spline. Lower MSE is better

We conduct a similar study to understand the impact of missing data fractions on classification tasks with MCAR masking scheme (see Figure 5). Similar to forecasting, we see, on average, a gradual decrease in the F1 scores with increasing missing fractions of the imputation-based approaches. We also observe a high range of variability in the Spline-imputed baselines, which suggests that the polynomial order of spline imputation can be further fine-tuned specific to the data. On the other hand, MissTSM shows consistently strong performance across all the three datasets.

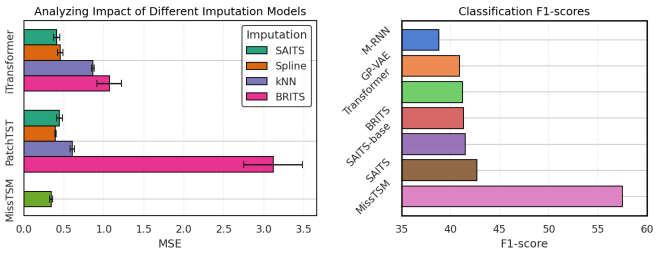
5.1.2 Impact of the Nature of Missing Values.

To understand the performance of imputation-based and imputation-free approaches under varying conditions of missing data, we compared their results across the two synthetic masking patterns: MCAR and Periodic Masking. From Fig. 6, we can observe that for the ETTh2 dataset, models perform consistently better under random masking compared to periodic. We can also see that the performance difference between MCAR and Periodic masking is, on an average, higher for SAITS-imputed models compared to Spline. This suggests that the hyper-parameters of SAITS can be further fine-tuned on the Periodic dataset, which is relatively easier for Spline to model. Additionally, the performance under MCAR and Periodic missingness on Weather dataset is comparatively similar, which hint towards high seasonality within the weather dataset, thus helping the imputation-based baselines on this dataset.

5.1.3 Impact of Imputation Methods.

The choice of imputation method dictates the overall performance of imputation-based frameworks. In Figure 7a, we compare four imputation techniques: Spline, kNN, BRITS, and SAITS, paired with two MTS models (iTransformer and PatchTST) at a forecasting horizon of $T = 720$ and 70% missing data fraction. Model performance on BRITS-imputed data is relatively poor, whereas models trained on SAITS-imputed data performs relatively good. This difference in performance indicates the impact of imputation models on downstream tasks within imputation-based frameworks. Notably, MissTSM-based imputation-free model achieves relatively low MSE scores compared to most imputation-based frameworks.

In Figure 7b, we compare MissTSM with six imputation baselines— M-RNN (Yoon et al., 2018b), GP-VAE (Fortuin et al., 2020), BRITS (Cao et al., 2018), Transformer (Vaswani et al., 2017), and SAITS (Du et al., 2023) - on a popular real-world classification dataset, PhysioNet (Silva et al., 2012) following the same evaluation setup as proposed in (Du et al., 2023). MissTSM achieves an impressive F1-score of 57.84%, representing an approximately 15% improvement over the best-performing model (trained on SAITS imputed data). This substantial performance gain on a real-world dataset with missing values highlights the potential of imputation-free or single-stage approaches compared to imputation-based approaches.



(a) Varying imputation models. Performance on ETTh2.

(b) Classification results of imputation models on PhysioNet.

Figure 7: Comparison of MSE (lower is better) and F1-score across imputation methods.

5.2 Comparing Model-Agnostic vs. Specialized Models

5.2.1 Analyzing MissTSM on IMTS Classification and Forecasting

We evaluate MissTSM on both classification and forecasting tasks for irregular multivariate time series. To illustrate the generality of our approach, we study two case models: (i) GRU-D, a specialized classifier for irregularly sampled data, and (ii) Latent ODE, a continuous-time generative model not originally designed for forecasting but adapted here to a long-term prediction setting. These case studies emphasize how specialized methods struggle when moved beyond their intended use, underscoring the value of model-agnostic approaches.

IMTS Classification. We conduct experiments on the IMTS classification task using the P12 (Goldberger et al., 2000b) and P19 (Reyna et al., 2020) datasets, following the same evaluation protocol as Luo et al. (2025). We report the baseline results for the considered models directly from Luo et al. (2025). Table 1 highlights the strong potential of model-agnostic approaches; integrating the MissTSM layer, can achieve performance on par with or exceeding that of several well-known IMTS models.

Table 1: Performance comparison on P19 and P12. Best in bold, second-best underlined.

Methods	P19		P12	
	AUROC	AUPRC	AUROC	AUPRC
GRU-D	<u>88.7</u> 1.2	<u>56.2</u> 2.3	79.6 0.6	41.7 1.8
ODE-RNN	87.1 1.0	52.6 3.2	78.8 0.6	37.4 2.6
SeFT	84.0 0.3	49.3 0.5	78.1 0.5	35.9 0.8
mTAND	82.9 0.9	32.2 1.5	**85.3** 0.3	**49.3** 1.0
Raindrop	87.6 2.7	**61.1** 1.4	82.0 0.6	42.7 1.7
MTGNN	88.5 1.0	55.8 1.5	82.1 1.5	41.8 2.1
MissTSM	**88.8** 1.3	<u>56.5</u> 1.2	<u>82.2</u> 0.5	<u>43.8</u> 1.1

Table 2: Comparing (F1 scores) MissTSM approach against GRU-D for classification datasets.

Dataset	GRU-D	MissTSM
Epilepsy	6.52%	64.9%
Gesture	3.16%	55.70%
EMG	2.78%	59.45%

Table 3: Comparing (MSE values) MissTSM with Latent ODE adapted for forecasting

Fraction	Latent ODE	MissTSM
60%	4.25	0.243
70%	3.181	0.250
80%	2.543	0.264
90%	2.624	0.316

Comparing MissTSM with GRU-D on Classification. To analyze the potential of model-agnostic approaches we apply the same MissTSM-integrated MAE model on synthetically masked (80%) classification datasets and compare against GRU-D. From Table 2, we observe that while GRU-D is a specialized model for IMTS data, the proposed model-agnostic still outperforms it significantly. Please refer to the Appendix for more implementation details.

Comparing MissTSM with Latent ODE on ETTh2. As discussed above, specialized IMTS models cannot be easily adapted to a different task. To analyze this further, we adapt the Latent ODE model (with ODE-RNN encoder) for a long-term forecasting problem and compare it against our model-agnostic approach. We consider a simple setup with 336 context length and 96 prediction length under MCAR masking with varying fractions. From Table 3, we see that Latent ODE struggles to perform long-sequence modeling, with significantly high MSE values. Moreover, ODE-based methods incur considerable computational costs, which grow even more pronounced for long-term modeling.

5.2.2 Analyzing Model-Agnostic Nature of MissTSM.

To further analyze model-agnostic capability of the proposed approach we integrate MissTSM with other MTS models like PatchTST and iTransformer. Tables 4 and 5 show competitive performance of MissTSM integrated with PatchTST, revealing potential for plugging MissTSM with advanced MTS models for improved performance on downstream tasks even in the presence of missing values with minimal change to the MTS model architecture. Please refer to appendix for additional results.

Table 4: MSE (mean_{std}) for PatchTST with MissTSM under 60% masking.

Dataset	Horizon Window	PatchTST + MissTSM	PatchTST + SAITS	PatchTST + Spline
ETTh2	96	$\mathbf{0.317}_{0.004}$	$0.503_{0.013}$	$\underline{0.324}_{0.013}$
	192	$\mathbf{0.377}_{0.009}$	$0.512_{0.011}$	$\underline{0.399}_{0.017}$
	336	$\mathbf{0.380}_{0.011}$	$\underline{0.410}_{0.012}$	$0.431_{0.005}$
	720	$0.514_{0.033}$	$\mathbf{0.411}_{0.002}$	$\underline{0.436}_{0.017}$
ETTm2	96	$\underline{0.202}_{0.005}$	$0.322_{0.045}$	$\mathbf{0.169}_{0.000}$
	192	$\underline{0.261}_{0.002}$	$0.359_{0.036}$	$\mathbf{0.227}_{0.000}$
	336	$\underline{0.313}_{0.001}$	$0.408_{0.043}$	$\mathbf{0.285}_{0.001}$
	720	$\underline{0.420}_{0.027}$	$0.459_{0.035}$	$\mathbf{0.376}_{0.001}$
Weather	96	$\underline{0.206}_{0.014}$	$\mathbf{0.169}_{0.001}$	$0.270_{0.110}$
	192	$\underline{0.276}_{0.027}$	$\mathbf{0.212}_{0.000}$	$0.287_{0.080}$
	336	$\underline{0.309}_{0.024}$	$\mathbf{0.263}_{0.001}$	$0.325_{0.065}$
	720	$\underline{0.340}_{0.003}$	$\mathbf{0.333}_{0.001}$	$0.391_{0.060}$

Table 5: MSE (mean_{std}) for PatchTST with MissTSM under 70% masking.

Dataset	Horizon Window	PatchTST + MissTSM	PatchTST + SAITS	PatchTST + Spline
ETTh2	96	$\underline{0.322}_{0.004}$	$0.548_{0.050}$	$\mathbf{0.317}_{0.009}$
	192	$\underline{0.382}_{0.011}$	$0.561_{0.056}$	$\mathbf{0.380}_{0.005}$
	336	$\underline{0.384}_{0.008}$	$0.468_{0.059}$	$\mathbf{0.372}_{0.007}$
	720	$0.621_{0.025}$	$\underline{0.497}_{0.085}$	$\mathbf{0.419}_{0.015}$
ETTm2	96	$\underline{0.213}_{0.006}$	$0.405_{0.079}$	$\mathbf{0.177}_{0.009}$
	192	$\underline{0.266}_{0.003}$	$0.447_{0.086}$	$\mathbf{0.236}_{0.009}$
	336	$\underline{0.315}_{0.004}$	$0.494_{0.095}$	$\mathbf{0.293}_{0.007}$
	720	$\underline{0.432}_{0.025}$	$0.529_{0.092}$	$\mathbf{0.386}_{0.009}$
Weather	96	$\underline{0.204}_{0.014}$	$\mathbf{0.166}_{0.013}$	$0.262_{0.122}$
	192	$\underline{0.249}_{0.030}$	$\mathbf{0.207}_{0.009}$	$0.279_{0.092}$
	336	$\underline{0.304}_{0.026}$	$\mathbf{0.257}_{0.007}$	$0.317_{0.075}$
	720	$\underline{0.358}_{0.012}$	$\mathbf{0.329}_{0.008}$	$0.383_{0.071}$

5.2.3 Impact on Real-world Datasets

We observed that existing benchmark datasets used for forecasting represent a certain level of seasonality which makes it easier for imputation-based models to show adequate performance. However, in many real-world datasets such as those encountered in ecology, there are complex forms of temporal structure in the data beyond simple seasonality. We compare the performance of MissTSM integrated with two MTS models,

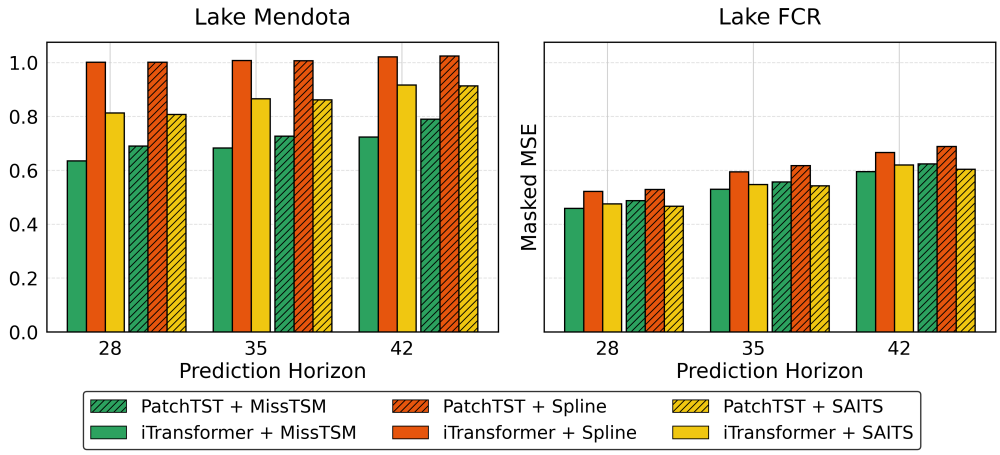


Figure 8: Forecasting performance comparison on Lake datasets across different prediction horizon windows. Lower MSE (y-axis) is better

iTransformer and PatchTST, on two Lake Datasets: Falling Creeks Reservoir and Mendota. Figure 8 reports masked MSE - MSE computed only on observed points - comparing MissTSM against imputation-based baselines. Competitive performance shown by MissTSM on both the real-world missing datasets further motivates the idea of imputation-free and model-agnostic approaches for IMTS modeling.

5.3 Ablations

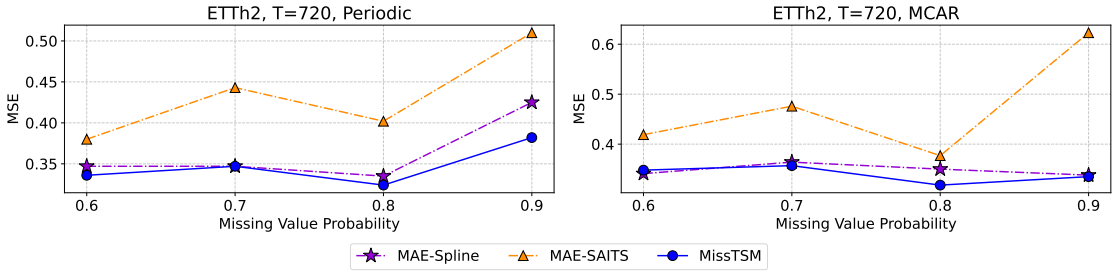


Figure 9: Ablations of MissTSM with and without TFI+MFAA layer on Forecasting datasets.

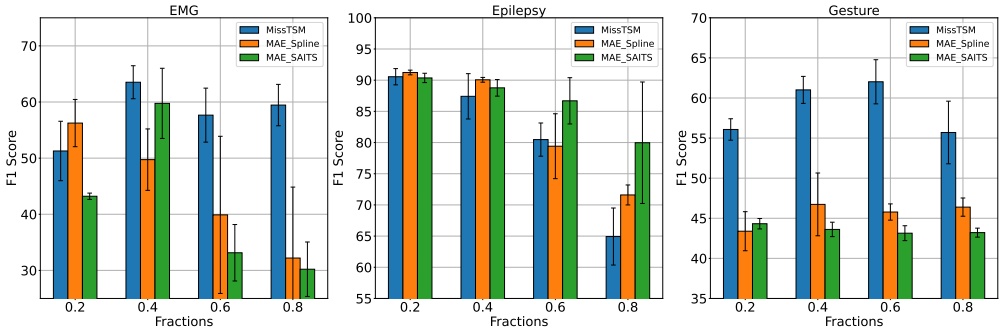


Figure 10: Ablations of MissTSM with and without the TFI+MFAA layer on the classification tasks.

In the ablation experiments, we evaluate the impact of integrating the MissTSM layer. We compare MAE with MissTSM against standard MAE (without MissTSM), using spline and SAITS imputation as additional baselines. The goal here is to understand the additional value of adding the MissTSM layer instead of

modeling on imputed data. For forecasting (Fig. 9) and classification (Fig. 10), MissTSM consistently improves performance. In forecasting, MissTSM-MAE outperforms all MAE variants, while in classification, it is consistently comparable or superior across all three datasets.

6 Conclusion

We investigate the performance of existing IMTS models as well as our proposed MissTSM framework on a variety of datasets and tasks with varying conditions of missing values. We show that imputation-based frameworks built on simple imputations perform well when the amount of missingness is small or there is periodic structure in the data (e.g., in Weather data) that is easy to approximate. However, imputation-based approaches show poor performance at larger missing value fractions and when missing values have limited periodic patterns (e.g., on the lake datasets). We also show that MissTSM, which is an imputation-free and model-agnostic framework shows competitive performance across most datasets, tasks, and settings compared to imputation-based and existing imputation-free specialized models. We hope our findings could inspire further research into developing flexible, model-agnostic adapters for handling the challenges in irregularly-sampled time-series data.

Limitations and Future Directions. (1) A limitation of the MFAA layer is that it doesn't learn the non-linear temporal dynamics and relies on the subsequent transformer encoder blocks to learn the dynamics. Future work can explore modifications of the MFAA layer such that it can jointly learn the cross-channel correlations with the non-linear temporal dynamics. (2) Independent embedding of each time-feature token can become computationally expensive in high-dimensional multivariate systems.

Broader Impact Statement

Validation requirements and safety considerations when used in healthcare sectors MissTSM is a decision-support tool and would require rigorous clinical validation to ensure reliable predictions. We would recommend that the model must be used in an "expert-in-the-loop" system

Guidance on when the method is reliable vs when it may fail The method is reliable in challenging scenarios with low periodicity, high missingness and high number of variates. On low variate, highly periodic data, a simpler approach like Spline + Backbone can be more effective

The potential issue of over-trust in imputation-free outputs We agree that this is a critical concern. Our recommendation would be to pair MissTSM's predictions with a reliable uncertainty quantification metric. This metric should reflect the sparsity (or quality) of the input data, warning the user when the predictions are of low confidence.

References

Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, 15–18 July 2004*, pp. 639–647. Springer, 2004.

Hyun Ahn, Kyunghee Sun, and Kwanghoon Pio Kim. Comparison of missing data imputation methods in time series forecasting. *Computers, Materials & Continua*, 70(1):767–779, 2022.

Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

Gustavo EAPA Batista, Maria Carolina Monard, et al. A study of k-nearest neighbour as an imputation method. *His*, 87(251-260):48, 2002.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

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

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Multivariate time series imputation by graph neural networks. corr abs/2108.00298 (2021). *arXiv preprint arXiv:2108.00298*, 2021.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.

Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 459–469, 2023.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pp. 1651–1661. PMLR, 2020.

David S Fung. Methods for the estimation of missing values in time series. 2006.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000a.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000b.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *International Conference on Machine Learning*, pp. 4353–4363. PMLR, 2020.

Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023.

Zachary C. Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. *Machine learning for healthcare conference*, pp. 253–270, 2016.

Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

Shuai Liu, Xiucheng Li, Gao Cong, Yile Chen, and Yue Jiang. Multivariate time-series imputation with disentangled temporal representations. In *The Eleventh International Conference on Learning Representations*, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems*, 32, 2019.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch gnn for irregular multivariate time series. In *Forty-second International Conference on Machine Learning*, 2025.

John J Magnuson, Stephen R Carpenter, and Emily H Stanley. North Temperate Lakes LTER: High Frequency Data: Meteorological, Dissolved Oxygen, Chlorophyll, Phycocyanin - Lake Mendota Buoy 2006 - current ver 39. Environmental Data Initiative, 2024. URL `https://portal.edirepository.org/nis/mapbrowse?packageid=knb-lter-ntl.129.39`.

Benjamin M Marlin, David C Kale, Robinder G Khemani, and Randall C Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*, pp. 389–398, 2012.

Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1):1049–1060, 1998.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022a.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022b.

Matthew A Reyna, Christopher S Josef, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Shamim Nemati, Gari D Clifford, and Ashish Sharma. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. *Critical care medicine*, 48(2):210–217, 2020.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Satya Narayan Shukla and Benjamin M. Marlin. A survey on principles, models and methods for learning from irregularly sampled time series. *arXiv preprint arXiv:2012.00168 (2020*, 2020.

Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318*, 2021.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pp. 245–248. IEEE, 2012.

P.N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to Data Mining*. What's New in Computer Science Series. Pearson, 2019. ISBN 9780133128901. URL https://books.google.com/books?id=_ZQ4MQEACAAJ.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34: 24804–24816, 2021a.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34: 24804–24816, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Weather. https://www.bgc-jena.mpg.de/wetter/, 2021.

Yuxi Wei, Juntong Peng, Tong He, Chenxin Xu, Jian Zhang, Shirui Pan, and Siheng Chen. Compatible transformer for irregularly sampled multivariate time series. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1409–1414. IEEE, 2023.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pp. 5689–5698. PMLR, 2018a.

Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018b.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. *arXiv preprint arXiv:2110.05357*, 2021.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in neural information processing systems*, 35, 2022.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.