Forecasting Irregularly Sampled Time Series with Transformer Encoders

Riccardo Benassi (✉), Francesco Del Buono, Giacomo Guiduzzi, and Francesco Guerra

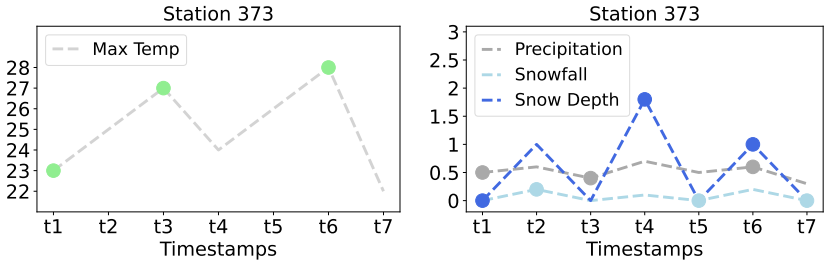
Università di Modena e Reggio Emilia, Italy
fistname.lastname@unimore.it

Abstract. Time series forecasting is a fundamental task in various domains, including environmental monitoring, finance, and healthcare. State-of-the-art forecasting models typically assume that time series are uniformly sampled. However, in real-world scenarios, data is often collected at irregular intervals and with missing values, due to sensor failures or network issues. This makes traditional forecasting approaches unsuitable. In this paper, we introduce `ISTF` (Irregular Sequence Transformer Forecasting), a novel transformer-based architecture designed for forecasting irregularly sampled multivariate time series (MTS). `ISTF` leverages exogenous variables as contextual information to enhance the prediction of a single target variable. The architecture first regularizes the MTS on a fixed temporal scale, keeping track of missing values. Then, a dedicated embedding strategy, based on a local and global attention mechanism, aims at capturing dependencies between timestamps, sources and missing values. We evaluate `ISTF` on two real-world datasets, FrenchPiezo and USHCN. The experimental results demonstrate that `ISTF` outperforms competing approaches in forecasting accuracy while remaining computationally efficient.

1 Introduction

A wide range of real-world phenomena across various domains, such as environmental monitoring, finance, and healthcare, can be naturally represented as time series. Advances in sensor technology, big data collection, data cleaning, and the Internet of Things have made it increasingly accessible for practitioners to acquire and manage such data. In particular, multivariate time series (MTS) are commonly used to consolidate temporal data from multiple sources, capturing different aspects of the same phenomenon (e.g., measurements from diverse sensors or the inclusion of exogenous variables).

The availability of many reliable datasets describing such real-world scenarios has also been the driver of technological innovations in the field of time-series analytics. Many approaches have been proposed as recently published surveys demonstrate [9, 18]. In particular, forecasting has been the recent focus of many innovations (e.g., see [9, 7, 15]) thanks to the application of the results achieved in the field of Machine Learning and Deep Learning. The resulting state-of-the-art approaches can accurately describe temporal information and incorporate



(a) Max temperature at Station 373. (b) Exogeneous data at Station 373.

Fig. 1: Motivating scenario: time series generated by sensors at Station 373. Notice how stations collect data at a different frequency and at different timestamps.

exogenous data to improve the forecast. However, they generally share the same requirement: time series should represent their observations with regular and uniform timestamps. Most of the approaches therefore assume that the data is collected on uniform intervals and that all the time series describing the phenomenon have the same pace. This can definitely be considered as an unrealistic condition in real-world scenarios, where sensors collect data with different intervals, paces and values may be missing in some timestamps due to failures of the network or the devices.

Motivating Scenario. Let us suppose that we need to forecast the maximum temperature recorded at monitoring station 373. The station has sensors that record both the time series of the maximum temperature (Figure 1a) and other exogenous data (Figure 1b). The goal is to forecast the maximum temperature of the next timestamps. We observe that the stations collect data with different frequencies and in different intervals, making it impossible to directly apply traditional forecasting techniques to this scenario.

Forecasting techniques for irregular time series broadly fall into two categories. The first relies on data imputation during preprocessing, to regularize the series before applying standard models.The second includes models natively designed for irregular data, which recognize irregularities and missing values and treat them as additional information [12, 2, 5, 17].

In this paper, we propose `ISTF` (Irregular Sequence Transformer Forecasting), an innovative architecture for the forecasting of irregularly sampled MTS that relies on contextual knowledge, provided by (1) other time stamps in the MTS; and (2) exogenous data sources. The architecture, described in Section 3, is built upon four main components: the Input Generator, responsible for extracting the relevant time series from the dataset; the Embedder and the Encoder, which generate embeddings for both the target and exogenous series; and the Forecaster, which produces the final predictions.

The approach has been experimentally evaluated against the real scenarios offered by the FrenchPiezo and USHCN datasets, regarding the water piezo-

metric levels in France and climate data in the USA respectively. The results demonstrate that `ISTF` outperforms competing approaches in effectiveness while remaining computationally feasible. In particular, training time is higher than the baselines, reaching up to an order of magnitude more, but remains manageable, with a maximum of six hours in the slowest configuration. Serving time, however, is in line with the baselines. The main contributions of the approach are: 1) the design of a transformer-based encoder architecture for forecasting that can manage forms of contributions in the prediction from exogeneity neighboring; 2) the experimentation of an technique that masks irregularly sampled MTS; 3) a deep experimentation on two large datasets. The code of `ISTF` and the experiments presented in the paper are available in the project GitHub[1].

2 Background

2.1 Related Work

Forecasting in time series is a long-standing research problem [6]. Traditional approaches are based on probability and statistics. More recently, approaches based on machine learning and deep learning have demonstrated to achieve great accuracy levels [9]. In particular, several works have successfully applied and extended transformers-based architecture to deal with time series analytic. [18] reviews the proposed variants of transformers for modeling time series data. The main modifications include enhancement in the positional encoding, in the attention module and in the architecture.

The majority of existing forecasting techniques cannot deal with irregular time series. Typically, data preprocessing is required before of their application [9]. The field of irregular time series forecasting has experienced significant advancements in recent years, with researchers exploring a variety of methodologies to handle missing values and irregular sampling intervals.

Usually, missing values in time series are addressed through heuristic or unsupervised imputation methods. Common practices [16] include omitting missing data, smoothing, interpolation, and spline methods. However, these techniques often fail to capture variable correlations and complex patterns, leading to suboptimal performance, especially in cases with high rates of missing data [4].

A paradigm shift occurred with [8], where absence is treated as a feature rather than an artifact to be corrected. The paper demonstrates that this kind of strategy significantly enhances predictive performance, particularly in classifying diagnoses with clinical time series data. Several approaches proposing a similar idea have been proposed. Among them, we selected GRU-D [2], mTAN [14], InterpNet [13], and PrimeNet [3] as representative approaches to be used as baselines in the evaluation of our proposal. GRU-D is a deep learning approach that effectively utilizes missing patterns in time series data. By incorporating masking and time intervals into a Gated Recurrent Unit (GRU) framework,

[1] https://github.com/softlab-unimore/ISTF

GRU-D is able to capture long-term temporal dependencies and utilize informative absence patterns for improving prediction accuracy. `ISTF` relies on masking and positional encoding to deal with irregularity, too. The main difference is the model architecture, a variant of a transformer-based model for `ISTF`. mTAN is based on a transformer architecture[14]. The key innovation here is using time embedding as both queries and keys in the attention mechanism, allowing the model to attend to observations at different time points. `ISTF` differs from mTAN at the architecture level. They are both based on a transformer, but `ISTF` relies on three modules to manage the contribution of exogenous data.

Other interesting DL-based approaches are InterpNet [13] and PrimeNet [3]. InterpNet is a deep learning architecture that combines a semi-parametric interpolation network with a prediction network, allowing information sharing across multiple dimensions during the interpolation stage. PrimeNet is a self-supervised learning framework that utilizes time-sensitive contrastive learning and data reconstruction task.

In summary, irregular time series forecasting has evolved from simple imputation methods to sophisticated deep learning models that effectively leverage the information in absence patterns and irregular sampling intervals.

2.2 Problem Definition

Let us start from the formalization in [17], where $\mathcal{D} = \{m_1, m_2, \ldots, m_N\}$ is a dataset of N MTS. Each MTS is defined as a sequence of observations collected over time, called features or variables, in the form of irregularly sampled univariate time series $\mathbf{m_i} = \{m_{i,1}, m_{i,2}, \ldots, m_{i,F_i}\}$, with F_i the dimension of the MTS m_i. We denote with N_{ij} as the number of data of the j-th univariate time series of m_i. The univariate time series can be represented as $m_{i,j} = [(t_{i,j,1}, x_{i,j,1}), (t_{i,j,2}, x_{i,j,2}), \ldots, (t_{i,j,N_{i,j}}, x_{i,j,N_{i,j}})]$, where $x_{i,j,k}$ is the value observed at time step k (i.e., at time $t_{i,j,k}$) of the j-th univariate time series of m_i. Since we are dealing with irregularly sampled MTS, different univariate time series may include a different number of observations collected in different times. This means that $N_{i,j} \neq N_{i,z}$ for $j \neq z$ and $t_{i,j,a} \neq t_{i,z,a}$ $\forall a, j \neq z$. We define as the **target series** the time series $m_{i,j}$, which represents the variable the user aims to forecast.

Finding Exogenous time series. We assume the existence of a function $findEx$ for the target series $m_{i,j}$ that selects from the MTS in the dataset $\mathcal{D}$ the exogeneous time series $E = \{e_1, \ldots, e_e\}$, where each e_k is a univariate time series:

$$E = findEx(m_{i,j}, \mathcal{D})$$

Forecasting. `ISTF` predicts the values of a given time series at time step $t + n$ using its past data and the exogenous time series.

$$\hat{y}_{t+n} = f(m_{i,j,1}, m_{i,j,2}, \ldots, m_{i,j,t}, e_{1,1}, \ldots, e_{1,t'}, e_{e,1}, \ldots, e_{e,t'}) \tag{1}$$

where the $m_{i,j,t}$, and $e_{u,t'}$ points are the historical data points for the target and the exogeneous series. We recall that time step t corresponds to the actual

timestamp $t_{i,j,t}$ of the univariate time series $m_{i,j}$. Since the timestamps of the series can correspond to different steps, we denote by t' all time steps associated with timestamps preceding t in the target series. To simplify the notation, we assume that all MTS have the same number of features ($F_i = F, \forall i$), and that the time steps are "normalized" for the MTS in the collection $\mathcal{D}$. This means that, the features of a MTS can assume the null value for the time steps that correspond to timestamps which are not sampled in the non-normalized series. Given a MTS m_i, we indicate with $M_i(t)$ the sequences of values for all the features composing m_i for the time steps $1, 2, \ldots, t$. The same notation is applied to exogenous ($E(t)$) series. With this simplification, Equation 1 can be reformulated in:

$$\hat{y}_{t+n} = f(m_{i,j}(t), e_1(t), \ldots, e_e(t))$$

2.3 Overview

The architecture of `ISTF` is designed around two key considerations: (1) real-world time series data is often irregular, with missing values and various sampling rates; and (2) real-world phenomena are typically described by multiple interdependent variables, such as temperature, humidity, and precipitation. As highlighted in Figure 2, `ISTF` expects two inputs: the target time series, which represents the phenomenon under investigation and whose future values need to be predicted, and a set of MTS exogenous series, which provide contextual environmental data.

The role of the *Embedder* component is to construct vector representations of the input time series. These embeddings provide a uniform representation of each timestamp of each variable as a fixed-size array, combining its value with information about its position in the series and its sampling date. `ISTF` implementation of the `ISTF` *Encoder* component enriches the representation provided by the embedder by incorporating knowledge of the interdependencies between the input time series. The `ISTF` *Encoder* extends the transformer encoder architecture with a local and global attention mechanism. In particular, local attention is the self-attention applied to each series separately, whereas global attention considers all points across all series. This way, the forecasting model (`ISTF` relies on a GRU model followed by a linear layer) can capture both series-specific patterns and inter-series relationships, which are crucial for accurate forecasting.

3 The `ISTF` Model

`ISTF` is conceived as a transformer model based architecture composed of 4 main components as represented in Figure 2: the Input Generator (in Section 3.1), the Embedder (in Section 3.2), the Encoder (in Section 3.3) and the Forecaster (in Section 3.4).

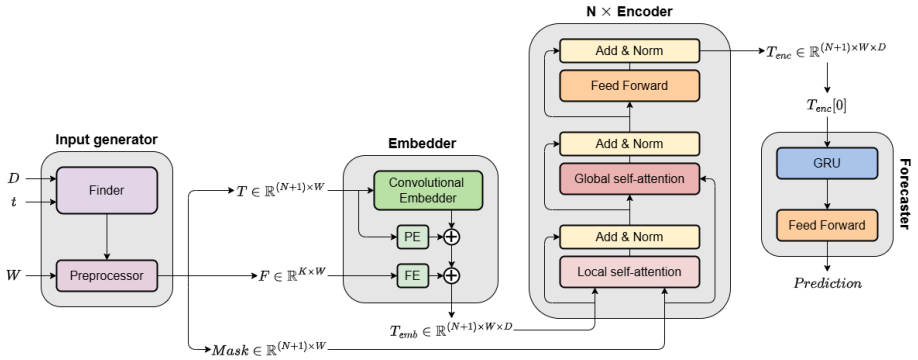


Fig. 2: `ISTF` Architecture

3.1 The Input Generator

The goal of the Input Generator component is to construct the input elements for `ISTF` from a target univariate time series and a multivariate dataset. The component consists of two modules: `Finder` implements the function $findEx$, which returns the n univariate time series in $\mathcal{D}$ that influence the forecasting of the target series t. Providing an implementation for $findEx$ is beyond the scope of this paper. A straightforward implementation is to select as exogenous series all signals in the MTS except for the target series. Moreover, domain knowledge can be used to identify which of these signals are actually relevant for the problem at hand. The preprocessor handles the irregular sampling of both the target and exogenous time series of interest. In particular, it addresses irregularities by aligning all series to the timestamps of the series with the highest granularity. One possible strategy to impute the missing values, which arise when aligning coarser time series to the finest resolution, is to use the last observed value prior to the missing timestamp. However, the approach is agnostic to the specific imputation method adopted. The component keeps track of the steps containing imputed values through a dedicated $Mask$ matrix and also maps the temporal features used by the original selected series into the ones with the highest granularity through the F matrix.

Finally, the *Input Generator* component computes three output matrices, considering the time window of interest W specified by the user. In particular:

- $T \in \mathbb{R}^{(N+1)\times W}$ represents the values of the target series (indexed by $T[0]$) and the exogenous series (indexed by $T[1:N]$) within the given time window.
- $F \in \mathbb{R}^{K\times W}$ contains the K temporal features extracted from the original $N+1$ univariate series.
- Mask $\in \mathbb{R}^{(N+1)\times W}$ is a boolean matrix indicating the imputed values (set to 1) during the "regularization" process of the series.

3.2 The Embedder

The goal of the ISTF Embedder is to standardize the heterogeneity of irregularly sampled input time series by generating a uniform embedding for each of them $t_{emb} \in T_{emb}$, with $T_{emb} \in \mathbb{R}^{(N+1)\times W\times D}$, where, according to the previous formalization, $N+1$ is the set of time series, W is the time window, and D is the dimension of the embedding, hyper-parameter of the approach.

The ISTF Embedder performs two main operations. Firstly, a Convolutional Neural Network (CNN) is applied to each series $i \in I$ to create an embedding vector $i_{emb} \in \mathbb{R}^{w\times d}$. Then, two kinds of positional encodings are added to the embeddings. The first, PE($\mathcal{P}$), with $\mathcal{P}$ the relative position associated to the time steps of i, is the positional encodings usually adopted in Transformer-based approaches. It assumes the form of Equation 2:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/D}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/D}}\right), \tag{2}$$

where *pos* is the relative position in the time series and D the dimension of the embedding. The second positional encoding is the variation introduced in [18, 21, 19] to incorporate the timestamp features (such as day, month, week, year, etc.), thus acknowledging the sequential and possible cyclical nature of the series. We call this PE($\mathcal{F}$), where $\mathcal{F}$ represents the timestamps (day, week, year,...) associated to the time steps of i through the matrix F.

The final results is the multi-faceted embedding $T_{emb} \in \mathbb{R}^{(N+1)\times W\times D}$ that encapsulates both the values of the time series and the temporal dynamics dictated by irregular sampling and missing values.

3.3 The Encoder

As in [5], the ISTF Encoder extends a vanilla transformer encoder with local attention, which captures intra-series relationships, and global attention, which models dependencies between target and exogenous series. The encoder is composed of a stack of identical layers, of which we describe a single one.

Local Attention. The intra-temporal dynamics of time series data are often complex, requiring advanced models to effectively capture the underlying patterns. The local attention mechanism is designed to learn these dynamics by generating embeddings for each time series independently, focusing exclusively on non-null values. This is achieved through the Mask matrix, which identifies missing values in the series, ensuring they are skipped during the self-attention computation. Lines 1–4 in Algorithm 1 describe the behavior of the local attention module, which applies a multi-head attention layer to the embeddings of each time series, properly masked.

Global Attention. ISTF computes new embeddings that represent inter-relations between the series, starting from the embeddings generated by the Local Attention component. Lines 5–9 in Algorithm 1 describe the procedure. First, the mask and the embeddings are reshaped to represent a bidimensional matrix, with time steps for each series in the rows and the embeddings in the columns.

Algorithm 1 The Encoder

Input: $T_{emb} \in \mathbb{R}^{(N+1)\times W\times D}$
Output: $T_{enc} \in \mathbb{R}^{(N+1)\times W\times D}$
`// Local attention`
1: **for** $i \in [0, ..., n]$ **do**
2: $I[i,:] = \text{MultiHeadAttention}(Q = T_{emb}[i], K = T_{emb}[i], V = T_{emb}[i], \text{Mask} = Mask[i])$
3: **end for**
4: $T_{local} = \text{norm}(T_{local} + T_{emb})$
`// Global attention`
5: $Mask\_global = \text{reshape}(Mask, (N+1, W) \rightarrow (N+1 \times W)$
6: $T_{local} = \text{reshape}(T_{local}, (N+1, W, D) \rightarrow (N+1 \times W, D))$
7: $T_{local} = \text{MultiHeadAttention}(Q = T_{local}, K = T_{local}, V = T_{local}, \text{mask} = Mask\_global)$
8: $T_{global} = \text{norm}(T_{global} + T_{local})$
9: $T_{global} = \text{reshape}(T_{global}, (V \times W, D) \rightarrow (V, W, D))$
`// Feedforward`
10: **for** $i \in [0, ..., n]$ **do**
11: $T_{enc}[i,:] = \text{FF}(T_{global}[i])$
12: **end for**
13: $T_{enc} = \text{norm}(T_{enc} + T_{global}) = 0$

Then, a Multi-Head Attention layer is applied to this data structure to generate the global attention embeddings.

Finally, the output embedding is obtained by applying a Feed-Forward layer to each embedding generated by the global attention module (lines 10–12), followed by a normalization layer that sums the global attention embeddings as residual components (line 13).

3.4 The Forecaster

The Encoder from the previous step generates an embedding for each timestamp of both the target series and the exogenous series, where each embedding has been related to the others. We preserve only the embeddings of the target series, which are then passed to a unidirectional GRU to obtain an aggregated representation of the embeddings:

$$f_E = GRU(T_{enc}[0,:]) \tag{3}$$

This is followed by a FeedForward Layer that computes the prediction

$$Prediction = FF(f_E) \tag{4}$$

4 Experimental Evaluation

The goal of the experimental evaluation is to answer the following research questions:

Table 1: Statistics about the Datasets used in the experiments.

Dataset	MTS	Points	Signals	Avg lenght	NaN percentage	Target NaN percentage
USHCN	1201	1744720	6	1453	3.97%	4.94%
FrenchPiezo	2664	6385608	4	2397	5.25%	12.1%

RQ1. **Effectiveness**. How accurate is `ISTF` architecture in making forecasts in scenarios with irregularly sampled time series data? (Section 4.2)
RQ2. **Ablation**. Are all components of `ISTF` architecture necessary, or can its complexity be reduced without sacrificing forecasting accuracy? (Section 4.3)
RQ3. **Robustness**. How sensitive is `ISTF` architecture to hyperparameter choices? (Section 4.4)
RQ4. **Efficiency**. How efficient is the `ISTF` architecture in terms of time performance? (Section 4.5)

4.1 Experimental Settings

Settings. The experiments have been performed on a Workstation with an NVIDIA L40S GPU with 48 GB of VRAM, 256 GB of RAM, and a dual AMD EPYC 9254 24-Core Processor. According to the literature in the field, the predictions are computed via single point regression [3]. Moreover, we used the hyperparameters defined for each baseline model as indicated in the original papers. For `ISTF`, we conducted experiments with the following configuration selected via a search on the validation set: a maximum of 100 training epochs combined with an early stopping patience of 20 epochs, a learning rate of $3 * 10^{-4}$, L2 regularization set to 10^{-2}, an embedding dimension of 32, 2 encoder layers and 4 attention heads. In all experiments, we employed the straightforward implementation of $FindEx$, which selects all signals in the MTS except for the target series as exogenous series. Each experiment was run three times with different random seeds, and the results were aggregated.

Datasets. We conducted experiments on two real MTS datasets, FrenchPiezo [10] and USHCN [11], which consist of irregularly sampled time series. Table 1 provides key statistics, including the number of MTS in each dataset, the total number of timestamps across all series, the number of signals in the MTS, the average series length, and the percentage of missing values in the dataset and in the target time series.

FrenchPiezo is a multivariate time series dataset from mainland France that monitors groundwater levels, also known as piezometric levels. It comprises 1,026 multivariate time series, each consisting of three dimensions: piezometric level (p), precipitation (tp), and evapotranspiration (e). Each series is associated with

a unique identifier (bss) corresponding to the piezometer that measures the piezometric level. The data, sampled daily from January 2015 to January 2021, span 2,221 days. The training period covers data from January 1, 2015, to January 1, 2020, and testing is conducted on data from January 1, 2020, to December 31, 2021. The objective is to forecast the piezometric levels.

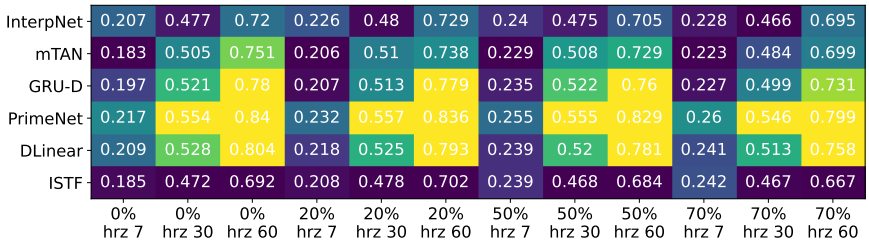
The United States Historical Climatology Network (*USHCN*) dataset includes daily records from 1,218 weather stations across the US, capturing six variables: precipitation, snowfall, snow depth, minimum temperature, maximum temperature, and average temperature. Each time series (TS) features irregular time intervals ranging from one to seven days, with varying sampling rates among them. The specific goal is to accurately forecast the average temperature for New York in the following days. Utilizing the cleaning procedure described in [1], we selected a subset of 1,168 meteorological stations, focusing on data spanning four years (1990 - 1993). The training dataset encompasses the years 1990 to 1992, and testing is performed on the year 1993 [12].

Baselines. We selected four approaches for irregular time series forecasting. GRU-D [2] is the common reference baseline, one of the earliest deep-learning approaches that handles missing data patterns. We also compare against more recent approaches: InterpNet [13], mTAN [14], and PrimeNet [3], which have achieved the highest results with irregular time series. Moreover, we include DLinear [20] as a representative of traditional forecasting approaches, which has also shown strong performance in regular time series tasks.

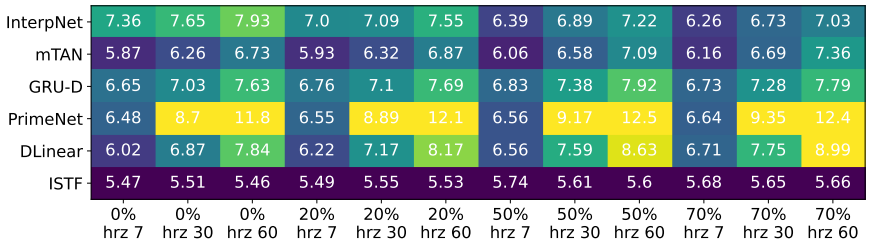
4.2 Forecasting accuracy

To evaluate the accuracy of `ISTF`, we performed multiple experiments using both the original datasets and modified versions where we artificially introduced missing values at rates of 20%, 50%, and 70% of the total data points. We then assessed its performance across different forecast horizons of 7, 30, and 60 days, using a fixed lookback window of 48 time steps. Figure 3 shows the results of the experiments: darker colors in the heatmap are associated to lower mean absolute error (MAE). Note that the Figure includes an experiment with 0% of missing value inserted. In this case the datasets still contain missing values as reported in Table 1. Table 2 reports the standard deviation only for the MAE due to space constraints, but similar trends are observed for the MSE.

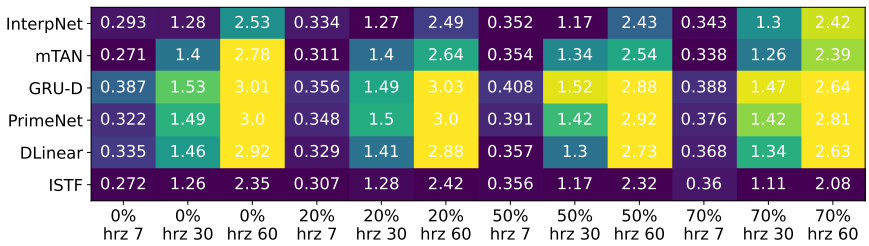
Discussion. The analysis of the results highlights two main aspects: (1) `ISTF` typically exhibits a lower error than other approaches. Figure 3 shows that our approach is the most effective for the majority of the dataset configurations in terms of both MAE and MSE. `ISTF` also tends to show a lower standard deviation compared to the other methods. (2) The error typically increases as the percentage of missing values and the forecasting horizon grow. While this trend holds for all approaches, the increase is less marked for `ISTF`, in particular in the USHCN dataset. (3) The standard deviation of `ISTF` is generally lower than that of the baselines, as reported in Table 2.



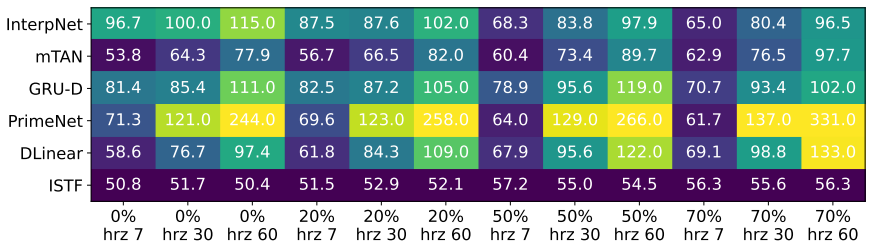
(a) FrenchPiezo (MAE)



(b) USHCN (MAE)



(c) FrenchPiezo (MSE)



(d) USHCN (MSE)

Fig. 3: Forecasting error measured using MAE and MSE. Darker colors in the heatmap indicate lower error. In the x-axis, the percentage represents the amount of inserted missing values and and *hrz* the forecast horizon.

Table 2: Standard deviation of MAE (similar for MSE).

	0% hrz 7	0% hrz 30	0% hrz 60	20% hrz 7	20% hrz 30	20% hrz 60	50% hrz 7	50% hrz 30	50% hrz 60	70% hrz 7	70% hrz 30	70% hrz 60
InterpNet	0.018	0.008	0.018	0.014	0.010	0.037	0.018	0.014	0.025	0.012	0.012	0.009
mTAN	0.006	0.010	0.005	0.018	0.010	0.019	0.014	0.033	0.035	0.006	0.013	0.014
GRU-D	0.012	0.020	0.024	0.005	0.010	0.015	0.004	0.007	0.022	0.004	0.007	0.016
PrimeNet	0.005	0.006	0.013	0.002	0.005	0.009	0.006	0.019	0.027	0.010	0.011	0.010
DLinear	0.006	0.007	0.011	0.003	0.007	0.006	0.007	0.013	0.021	0.002	0.010	0.008
ISTF	0.004	0.004	0.006	0.006	0.005	0.018	0.014	0.016	0.017	0.013	0.013	0.008

(a) FrenchPiezo (MAE)

	0% hrz 7	0% hrz 30	0% hrz 60	20% hrz 7	20% hrz 30	20% hrz 60	50% hrz 7	50% hrz 30	50% hrz 60	70% hrz 7	70% hrz 30	70% hrz 60
InterpNet	0.332	0.562	0.441	0.165	0.228	0.268	0.164	0.111	0.063	0.241	0.173	0.129
mTAN	0.330	0.386	0.430	0.166	0.181	0.272	0.068	0.118	0.041	0.236	0.216	0.115
GRU-D	0.368	0.415	0.487	0.210	0.242	0.321	0.031	0.126	0.085	0.206	0.170	0.095
PrimeNet	0.303	0.433	0.646	0.145	0.288	0.482	0.098	0.054	0.371	0.256	0.128	0.042
DLinear	0.293	0.346	0.437	0.137	0.221	0.285	0.093	0.029	0.040	0.259	0.204	0.104
ISTF	0.353	0.417	0.390	0.149	0.239	0.265	0.040	0.069	0.009	0.233	0.205	0.140

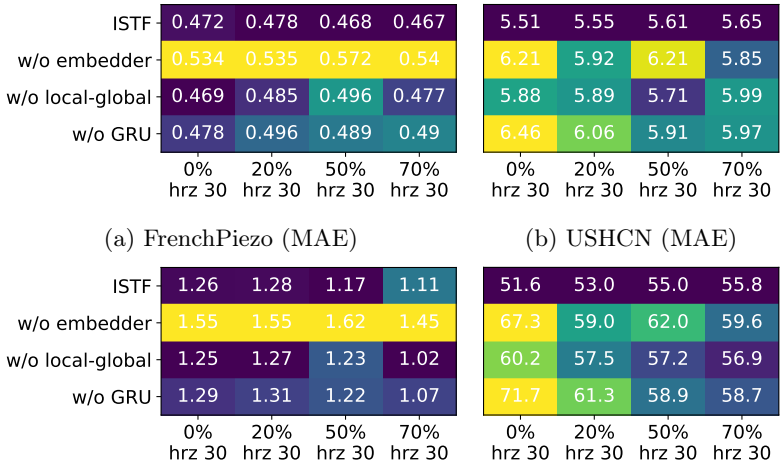
(b) USHCN (MAE)

Lesson learned. `ISTF` consistently outperforms other baselines across all experiments, demonstrating both superior forecasting accuracy and greater robustness to increasing missing values and longer forecasting horizons.

4.3 Ablation study

To evaluate the contribution of each component in the `ISTF` architecture, we performed an ablation study by systematically removing individual modules and analyzing their impact on forecasting accuracy. The complete model, without any modifications, serves as a baseline to assess the necessity of each component. The following ablation settings were considered:

1. **w/o Embedder**: The Embedder and positional encoder are removed, and the irregular time series are fed directly into the Encoder.
2. **w/o Local & Global Attention**: The attention mechanisms are replaced with a standard Transformer Encoder.



(a) FrenchPiezo (MAE) (b) USHCN (MAE)

(c) FrenchPiezo (MSE) (d) USHCN (MSE)

Fig. 4: Ablation study results. The rows represent the full model and its ablated versions; the columns correspond to different percentages of missing values for the FrenchPiezo and USHCN datasets. The cells report the error in terms of MAE and MSE.

3. **w/o GRU**: The Encoder output is directly used for forecasting, bypassing the GRU component.

Figure 4 presents the MAE and MSE errors for both datasets, considering missing value rates of 0%, 20%, 50% and 70%, and future horizon of 30 days. Table 3 shows the percentage error increase due to ablations.

Discussion. The experimental results clearly indicate that the primary contributor to error reduction is the Embedder. In particular, its removal leads to a 16% increase in MAE, averaged across missing value configurations for the FrenchPiezo dataset (8% for USHCN), whereas the removal of other components results in a more modest error increase of 3% (7% for USHCN). Similar considerations hold when evaluating the error using MSE.

Lesson learned. The results show that all components contribute to the model performance, but the Embedder is crucial for error reduction.

4.4 Robustness

To evaluate the sensitivity of `ISTF` to key hyperparameter changes and to identify the optimal configuration, we conducted four types of experiments on both the USHCN and FrenchPiezo datasets, as shown in Figure 5. We experimented with a missing value percentage of 0%, 20%, 50% and 50%, a prediction horizon of 30 days, and a look-back window of 48 time steps, varying the following parameters:

	FrenchPiezo					USHCN				
Model	0%	20%	50%	70%	Mean	0%	20%	50%	70%	Mean
w/o embedder	13.31%	11.91%	22.07%	15.66%	15.74%	12.62%	6.54%	10.79%	3.54%	8.37%
w/o local-global	-0.47%	1.47%	5.90%	2.23%	2.28%	6.68%	6.02%	1.78%	5.99%	5.11%
w/o GRU	1.38%	3.67%	4.49%	4.99%	3.63%	17.22%	9.07%	5.29%	5.75%	9.33%

(a) MAE

	FrenchPiezo					USHCN				
Model	0%	20%	50%	70%	Mean	0%	20%	50%	70%	Mean
w/o embedder	23.13%	21.07%	38.60%	29.97%	28.10%	30.45%	11.34%	12.88%	6.69%	15.34%
w/o local-global	-0.35%	-0.44%	5.40%	-7.97%	-0.84%	16.59%	8.36%	4.12%	1.89%	7.74%
w/o GRU	2.26%	2.91%	5.01%	-3.93%	1.56%	38.93%	15.55%	7.19%	5.12%	16.70%

(b) MSE

Table 3: Error increase due to ablations.

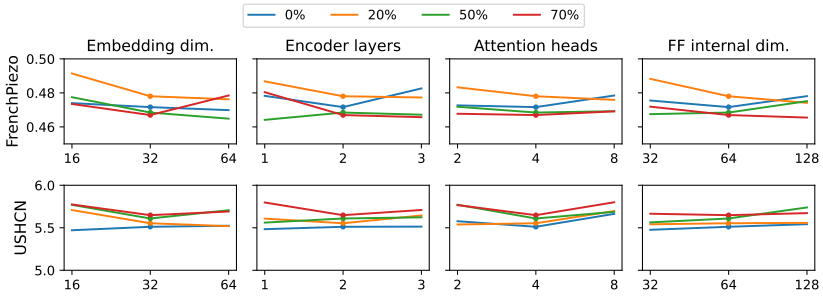
1. **Attention Heads:** we tested configurations with 2, 4 (default), and 8 heads to analyze the impact of the attention mechanisms.
2. **Embedding Dimension:** we examined embedding sizes of 16, 32 (default), and 64 dimensions to assess their effect on representation capacity. The GRU hidden size was set to match the embedding dimension in each respective configuration.
3. **Encoder Layers:** we explored architectures with 1, 2 (default), and 3 transformer encoder layers to measure the effect of depth on performance.
4. **Feed-Forward internal dimension**: we varied the internal dimension of the feed-forward network within each encoder layer, exploring values of 32, 64 (default), and 128.

Discussion. All experiments confirm that the selected hyperparameters minimize the error in both datasets. Moreover, the Figure shows that variations around the chosen values do not produce significant changes in forecasting accuracy, highlighting the robustness of the model to parameter tuning.

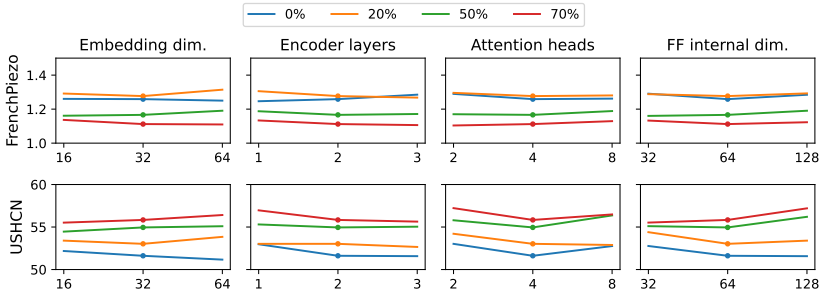
Lesson learned. `ISTF` shows robustness to hyperparameter variations, with minimal impact on performance.

4.5 Time Performance

We assessed the evaluation by measuring the time needed to train the datasets with 50% missing values. Table 4 shows the time required for training the



(a) MAE



(b) MSE

Fig. 5: Robustness to variations in hyperparameters. The x-axis represents the hyperparameter configurations; the y-axis the forecasting error (MAE and MSE).

datasets, reporting the total time, the time required to complete an epoch and a batch of 64 records.

Discussion. The experimental results show that ISTF is generally less efficient than the fastest baselines. However, its efficiency is comparable to GRU-D, and overall training times remain manageable, not hindering ISTF applicability in real-world scenarios. The longest training time is observed for the FrenchPiezo dataset, reaching two hours, close to the time recorded by InterpNet and slightly less than GRU-D, the slowest baseline overall. For the USHCN dataset, training takes slightly less than an hour, which is marginally slower than GRU-D, the least efficient baseline on this dataset. It is also worth noting that the time required to process a batch, which is relevant to assessing the serving time of `ISTF`, is limited and in line with the average of the other baselines.

Lesson learned. `ISTF` trades efficiency in training for improved accuracy, but training times remain feasible for real-world applications.

Table 4: Training time (seconds): total, per epoch, per batch.

Model	FrenchPiezo			USHCN		
	total	**epoch**	**batch**	**total**	**epoch**	**batch**
InterpNet	5304.155	160.732	0.022	1412.895	64.223	0.023
mTAN	4585.016	127.362	0.017	1505.682	48.57	0.018
GRU-D	8202.501	356.63	0.048	3166.397	143.927	0.052
PrimeNet	3963.591	74.785	0.010	869.753	28.992	0.010
DLinear	1037.824	24.710	0.003	464.587	9.481	0.003
`ISTF`	7461.781	226.115	0.030	3489.839	94.32	0.034

5 Conclusions

We proposed `ISTF`, a transformer-based model designed to handle irregularly sampled multivariate time series by integrating local and global attention mechanisms. Experimental results on two real-world datasets show that `ISTF` achieves superior forecasting accuracy compared to existing approaches. This improvement comes with higher computational time, but the overall cost remains manageable for real-world applications. The ablation study highlights the importance of every architecture component, confirming its role in reducing prediction errors. Future work will focus on optimizing efficiency and extending the model to broader forecasting scenarios.

Acknowledgments. This work was partially supported by the project AnomalyFeats - FARD2023 (Department of Engineering "Enzo Ferrari", UNIMORE, IT).

References

1. Brouwer, E.D., Simm, J., Arany, A., Moreau, Y.: Gru-ode-bayes: Continuous modeling of sporadically-observed time series. In: NeurIPS. pp. 7377–7388 (2019)
2. Che, Z., Purushotham, S., Cho, K., Sontag, D.A., Liu, Y.: Recurrent neural networks for multivariate time series with missing values. CoRR **abs/1606.01865** (2016)
3. Chowdhury, R.R., Li, J., Zhang, X., Hong, D., Gupta, R.K., Shang, J.: Primenet: Pre-training for irregular multivariate time series. In: AAAI. pp. 7184–7192. AAAI Press (2023)
4. Du, W., Wang, J., Qian, L., Yang, Y., Liu, F., Wang, Z., Ibrahim, Z.M., Liu, H., Zhao, Z., Zhou, Y., Wang, W., Ding, K., Liang, Y., Prakash, B.A., Wen, Q.: Tsi-bench: Benchmarking time series imputation. CoRR **abs/2406.12747** (2024)
5. Grigsby, J., Wang, Z., Qi, Y.: Long-range transformers for dynamic spatiotemporal forecasting. CoRR **abs/2109.12218** (2021)
6. Hyndman, R., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, Australia, 3rd edn. (2021)

7. Lim, B., Zohren, S.: Time series forecasting with deep learning: A survey. CoRR **abs/2004.13408** (2020)
8. Lipton, Z.C., Kale, D.C., Wetzel, R.C.: Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In: MLHC. JMLR Workshop and Conference Proceedings, vol. 56, pp. 253–270. JMLR.org (2016)
9. Liu, Z., Zhu, Z., Gao, J., Xu, C.: Forecast methods for time series data: A survey. IEEE Access **9**, 91896–91912 (2021)
10. Mbouopda, M.F., Guyet, T., Labroche, N., Henriot, A.: Experimental study of time series forecasting methods for groundwater level prediction. In: AALTD@ECML/PKDD. Lecture Notes in Computer Science, vol. 13812, pp. 34–49. Springer (2022)
11. Menne, M., Williams, Jr., C., Vose, R.: Long-term daily and monthly climate records from stations across the contiguous united states (u.s. historical climatology network) (1 2016). https://doi.org/10.3334/CDIAC/CLI.NDP019
12. Schirmer, M., Eltayeb, M., Lessmann, S., Rudolph, M.: Modeling irregular time series with continuous recurrent units. In: ICML. Proceedings of Machine Learning Research, vol. 162, pp. 19388–19405. PMLR (2022)
13. Shukla, S.N., Marlin, B.M.: Interpolation-prediction networks for irregularly sampled time series. In: ICLR (Poster). OpenReview.net (2019)
14. Shukla, S.N., Marlin, B.M.: Multi-time attention networks for irregularly sampled time series. In: ICLR. OpenReview.net (2021)
15. Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: A survey. Big Data **9**(1), 3–21 (2021)
16. Wang, J., Du, W., Cao, W., Zhang, K., Wang, W., Liang, Y., Wen, Q.: Deep learning for multivariate time series imputation: A survey. CoRR **abs/2402.04059** (2024)
17. Wang, Z., Zhang, Y., Jiang, A., Zhang, J., Li, Z., Gao, J., Li, K., Lu, C., Ren, Z.: Improving irregularly sampled time series learning with time-aware dual-attention memory-augmented networks. In: CIKM. pp. 3523–3527. ACM (2021)
18. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in time series: A survey. In: IJCAI. pp. 6778–6786. ijcai.org (2023)
19. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: NeurIPS. pp. 22419–22430 (2021)
20. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: AAAI. pp. 11121–11128. AAAI Press (2023)
21. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: AAAI. pp. 11106–11115. AAAI Press (2021)