FITS: Modeling Time Series with $10k$ Parameters

Xu Zhijian, Zeng Ailing, Xu Qiang
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
{zjxu21, qxu, alzeng}@cse.cuhk.edu.hk

Abstract

In this paper, we introduce FITS, a lightweight yet powerful model for time series analysis. Unlike existing models that directly process raw time-domain data, FITS operates on the principle that time series can be manipulated through interpolation in the complex frequency domain. By discarding high-frequency components with negligible impact on time series data, FITS achieves performance comparable to state-of-the-art models for time series forecasting and anomaly detection tasks, while having a remarkably compact size of only approximately $10k$ parameters. Such a lightweight model can be easily trained and deployed in edge devices, creating opportunities for various applications. The code is available in: https://github.com/VEWOXIC/FITS

1 Introduction

Time series analysis plays a crucial role in numerous domains, including finance, energy, weather forecasting, and signal processing, where understanding and predicting temporal patterns are essential. Existing time series analysis methods primarily focus on extracting features in the time domain (Zhou et al., 2021; Liu et al., 2022a; Zeng et al., 2022; Nie et al., 2023; Zhang et al., 2022). However, due to the inherent complexity and dynamic nature of time series data, the information contained in the time domain tends to be sparse and dispersed. Consequently, researchers design intricate methodologies and complex models to capture and exploit this information, often relying on approaches such as transformer architectures (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022a). However, these sophisticated techniques often lead to the proliferation of large-scale and computationally demanding models, posing challenges in terms of efficiency and scalability.

Conversely, the frequency domain representation of time series data offers a more concise and compact representation of its underlying information. Recognizing this potential, previous studies have explored the utilization of frequency domain information in time series analysis. For instance, FEDformer (Zhou et al., 2022a) incorporates spectral information as a supplementary feature, enhancing the modeling capabilities of transformer-based time series models. Another approach, FNet (Lee-Thorp et al., 2022), leverages frequency domain multiplication to replace convolution operations, thereby reducing computational overhead. Moreover, LTSF-Linear (Zeng et al., 2022) has demonstrated that highly accurate predictions can be achieved by solely learning the dominant periodicity. Similarly, methods like TimesNet (Wu et al., 2023) segment the time series based on frequencies with high amplitude and employ CNNs for multi-periodicity feature extraction.

However, existing methodologies often overlook the fundamental nature of the frequency domain representation, which utilizes complex numbers to express both amplitude and phase information. Motivated by the fact that longer time series segments provide a higher-resolution frequency representation, we propose FITS (**F**requency **I**nterpolation **T**ime **S**eries Analysis Baseline). The core component of FITS is a complex-valued linear layer that can explicitly learn amplitude scaling and phase shift to perform interpolation in the complex frequency domain. Although FITS conducts interpolation in the frequency domain, it remains an end-to-end time domain model incorporating the rFFT (Brigham & Morrow, 1967). Specifically, we project the input segment to the complex frequency domain for frequency interpolation using rFFT. We then project the interpolated frequency representation back to the time domain as a longer segment for supervision. This end-to-end design enables FITS to adapt to various downstream tasks with commonly-used time domain supervision, such as forecasting and reconstruction.

Additionally, FITS incorporates a low-pass filter to obtain a compact representation with minimal information loss, resulting in small model volume and minimal computational overhead while maintaining state-of-the-art (SOTA) performance. Notably, under most settings, FITS achieves SOTA performance with under **10k parameters**, which is **50 times smaller** than the lightweight temporal linear model DLinear (Zeng et al., 2022) and approximately **10,000 times smaller** than other mainstream models. The low memory and computation overhead make FITS suitable for deploying or even training on edge devices for forecasting or anomaly detection.

To summarize, our contributions are twofold:

- We introduce FITS, a lightweight model containing merely **5k∼10k** parameters for time series analysis. Despite its compact size which is several orders of magnitude smaller than mainstream models, FITS delivers exceptional performance in various tasks, including long-term forecasting and anomaly detection, achieving state-of-the-art performance in several datasets.
- FITS employs the complex-valued neural network for time series analysis, which provides a novel perspective that simultaneously captures amplitude and phase information, leading to more comprehensive and efficient modeling of time series data.

2 Related Work and Motivation

2.1 Frequency-aware Time Series Analysis Models

Recent advancements in time series analysis have witnessed the utilization of frequency domain information to capture and interpret underlying patterns. FNet (Lee-Thorp et al., 2022) leverages a pure attention-based architecture to efficiently capture temporal dependencies and patterns solely in the frequency domain, eliminating the need for convolutional or recurrent layers. On the other hand, FEDFormer (Zhou et al., 2022a) and FiLM (Zhou et al., 2022b) incorporate frequency information as supplementary features to enhance the model's capability in capturing long-term periodic patterns and speed up computation.

The other line of work aims to capture the periodicity inherent in the data. For instance, DLinear (Zeng et al., 2022) adopts a single linear layer to extract the dominant periodicity from the temporal domain and surpasses a range of deep feature extraction-based methods. More recently, TimesNet (Wu et al., 2023) achieves state-of-the-art results by identifying several dominant frequencies instead of relying on a single dominant periodicity. Specifically, they use the Fast Fourier Transform (FFT) to find the frequencies with the largest energy and reshape the original 1D time series into 2D images according to their periods.

However, these approaches still rely on feature engineering to identify the dominant period set. Selecting this set based on energy may only consider the dominant period and its harmonics, limiting the information captured. Moreover, these methodologies are still considered inefficient and prone to overfitting.

2.2 Divide and Conquer the Frequency Components

Treating a time series as a signal allows us to break it down into a linear combination of sinusoidal components without any information loss. Each component possesses a unique frequency, initial phase, and amplitude. Forecasting directly on the original time series can be challenging, but forecasting each frequency component is comparatively straightforward, as we only need to apply a phase bias to the sinusoidal wave based on the time shift. Subsequently, we linearly combine these shifted sinusoidal waves to obtain the forecasting result.

This approach effectively preserves the frequency characteristics of the given look-back window while maintaining semantic consistency between the look-back window and the forecasting horizon.

Specifically, the resulting forecasted values maintain the frequency features of the original time series with a reasonable time shift, ensuring that semantic consistency is maintained.

However, forecasting each sinusoidal component in the time domain can be cumbersome, as the sinusoidal components are treated as a sequence of data points. To address this, we propose conducting this manipulation in the complex frequency domain, which offers a more compact and information-rich representation, as described below.

3 Method

3.1 Preliminary: FFT and Complex Frequency Domain

The Fast Fourier Transform (FFT, (Brigham & Morrow, 1967)) is a widely used algorithm for efficiently computing the Discrete Fourier Transform (DFT) of a sequence of complex numbers. The DFT is a mathematical operation that converts a discrete-time signal from the time domain to the complex frequency domain. In cases where the input signal is real, such as in time series analysis, the Real FFT (rFFT) is commonly used to obtain a compact representation. With an input of N real numbers, the rFFT produces a sequence of $N/2+1$ complex numbers that represent the signal in the complex frequency domain.

Complex Frequency Domain

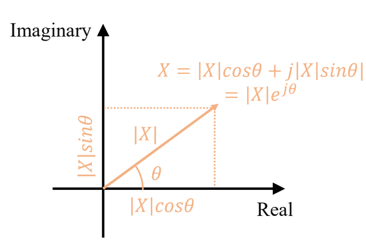
In Fourier analysis, the complex frequency domain is a representation of a signal in which each frequency component is characterized by a complex number. This complex number captures both the amplitude and phase of the component, providing a comprehensive description. The amplitude of a frequency component represents the magnitude or strength of that component in the original time-domain signal. In contrast, the phase represents the temporal shift or delay introduced by that component. Mathematically, the complex number associated with a frequency component can be represented as a complex exponential element with a given amplitude and phase:

$$X(f) = |X(f)|e^{j\theta(f)},$$

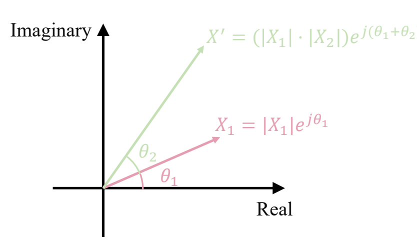
where $X(f)$ is the complex number associated with the frequency component at frequency f, $|X(f)|$ is the amplitude of the component, and $\theta(f)$ is the phase of the component. As shown in Fig. 1(a), in the complex plane, the complex exponential element can be visualized as a vector with a length equal to the amplitude and angle equal to the phase:

$$X(f) = |X(f)|(\cos\theta(f) + j\sin\theta(f))$$

Therefore, the complex number in the complex frequency domain provides a concise and elegant means of representing the amplitude and phase of each frequency component in the Fourier transform.



(a) Complex number on the complex plane



(b) Complex number multiplication

Figure 1: Illustration of Complex Number Visualization and Multiplication

Time Shift and Phase Shift. The time shift of a signal corresponds to the phase shift in the frequency domain. Especially in the complex frequency domain, we can express such phase shift by multiplying a unit complex exponential element with the corresponding phase. Mathematically, if we shift a signal $x(t)$ forward in time by a constant amount τ, resulting in the signal $x(t-\tau)$, the Fourier transform is given by:

$$X_\tau(f) = e^{-j2\pi f\tau}X(f) = |X(f)|e^{j(\theta(f)-2\pi f\tau)} = [cos(-2\pi f\tau) + jsin(-2\pi f\tau)]X(f)$$

The shifted signal still has an amplitude of $|X(f)|$, while the phase $\theta_\tau(f) = \theta(f) - 2\pi f\tau$ shows a shift which is linear to the time shift.

In summary, the amplitude scaling and phase shifting can be simultaneously expressed as the multiplication of complex numbers, as shown in Fig. 1(b).

3.2 FITS Pipeline

Motivated by the fact that a longer time series provides a higher frequency resolution in its frequency representation, we train FITS to generate an extended time series segment by interpolating the frequency representation of the input time series segment. We use a complex-valued linear layer to learn such interpolation. According to the fact that the amplitude scaling and phase shifting can be conveniently expressed as the multiplication of complex numbers, such complex linear combination allows FITS to effectively incorporate both the amplitude scaling and phase shift of frequency components during the interpolation process. As shown in Fig. 2, we use rFFT to project time series segments to the complex frequency domain. After the interpolation, the frequency representation is projected back with inverse rFFT (irFFT).

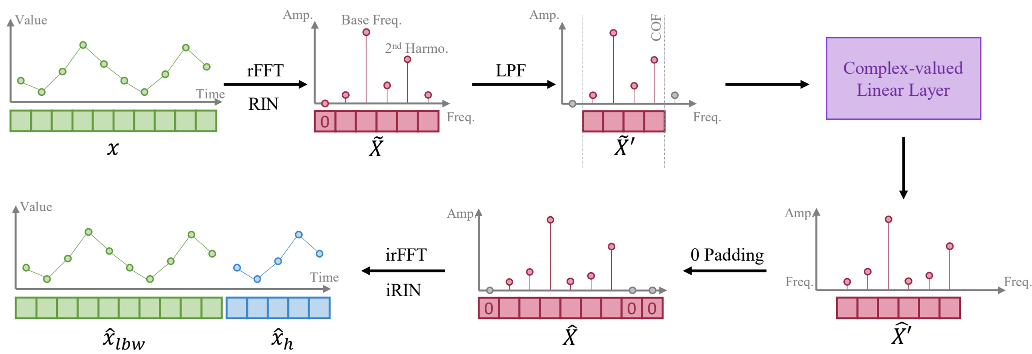


Figure 2: Pipeline of FITS, with a focus on the forecasting task. The reconstruction task follows the same pipeline, except for the reconstruction supervision loss.

However, we cannot directly use the frequency representation of the original input time series segment because the mean of such segments will result in a very large 0-frequency component in its complex frequency representation. To eliminate the 0-frequency component, we pass it through reversible instance-wise normalization (RIN) (Kim et al., 2022) to obtain a zero-mean instance. As a result, the normalized complex frequency representation now has a length of $N/2$, where N represents the original length of the time series.

Furthermore, we incorporate a low-pass filter (LPF) into the FITS model to further reduce its size. The LPF removes high-frequency components above a specified cutoff frequency, resulting in a more compact model representation while retaining the important information of the time series. The rationale behind this design will be elaborated in the subsequent section. Despite operating in the frequency domain, FITS is supervised in the time domain using common loss functions such as Mean Squared Error (MSE) after the irFFT, allowing for diverse supervision tailored to different time series downstream tasks.

In the case of forecasting tasks, we generate the look-back window along with the horizon as shown in Fig. 2. This allows us to provide supervision for forecasting and backcasting, where the model is encouraged to accurately reconstruct the look-back window. Our ablation study reveals that combining backcast and forecast supervision can yield improved performance in certain scenarios.

For reconstruction tasks, we downsample the original time series segment based on a specific downsampling rate. Subsequently, FITS is employed to perform frequency interpolation, enabling the reconstruction of the downsampled segment back to its original form. Thus, direct supervision is applied using reconstruction loss to ensure faithful reconstruction. The reconstruction tasks also follow the pipeline in Fig. 2 with the supervision replaced with reconstruction loss.

3.3 Key Mechanisms of FITS

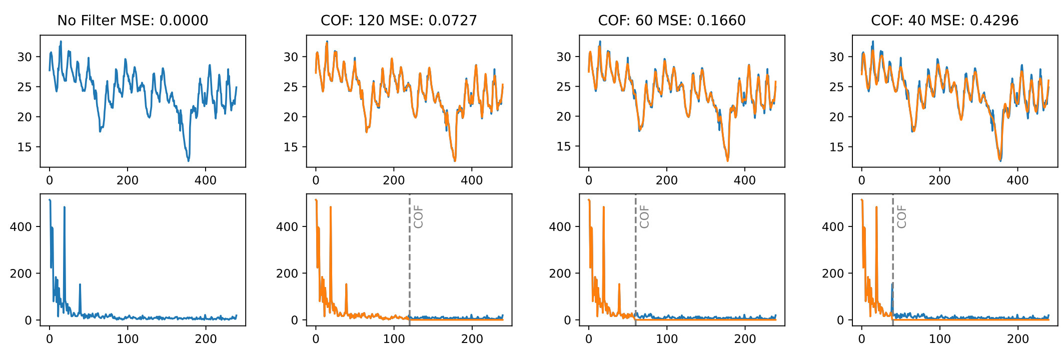
Complex Frequency Linear Interpolation. To control the output length of the model, we introduce an interpolation rate denoted as η, which represents the ratio of the model's output length L_o to its corresponding input length L_i.

It is worth noting that frequency interpolation operates on the normalized complex frequency representation, which has half the length of the original time series. Importantly, this interpolation rate can also be applied to the frequency domain, as indicated by the equation:

$$\eta_{freq} = \frac{L_o/2}{L_i/2} = \frac{L_o}{L_i} = \eta$$

Based on this formula, with an arbitrary frequency f, the frequency band $1 \sim f$ in the original signal is linearly projected to the frequency band $1 \sim \eta f$ in the output signal. As a result, we define the input length of our complex-valued linear layer as L and the interpolated output length as ηL. Notably, when applying the Low Pass Filter (LPF), the value of L corresponds to the cutoff frequency (COF) of the LPF. After performing frequency interpolation, the complex frequency representation is zero-padded to a length of $L_o/2$, where L_o represents the desired output length. Prior to applying the irFFT, an additional zero is introduced as the representation's zero-frequency component.

Low Pass Filter (LPF). The primary objective of incorporating the LPF within FITS is to compress the model's volume while preserving essential information. The LPF achieves this by discarding frequency components above a specified cutoff frequency (COF), resulting in a more concise frequency domain representation. The LPF retains the relevant information in the time series while discarding components beyond the model's learning capability. This ensures that a significant portion of the original time series' meaningful content is preserved. As demonstrated in Fig. 3, the filtered waveform exhibits minimal distortion even when only preserving a quarter of the original frequency domain representation. Furthermore, the high-frequency components filtered out by the LPF typically comprise noise and trends, which are inherently irrelevant for effective time series modeling.



(a) Original (b) COF at 6th harmonic (c) COF at 3rd harmonic (d) COF at 2nd harmonic

Figure 3: Waveform (1st row) and amplitude spectrum (2nd row) of a time series segment selected from the 'OT' channel of the ETTh1 dataset, spanning from the 1500th to the 1980th data point. The segment has a length of 480, and its dominant periodicity is 24, corresponding to a base frequency of 20. The blue lines represent the waveform/spectrum with no applied filter, while the orange lines represent the waveform/spectrum with the filter applied. The filter cutoff frequency is chosen based on a harmonic of the original time series.

Selecting an appropriate cutoff frequency (COF) remains a nontrivial challenge. To address this, we propose a method based on the harmonic content of the dominant frequency. Harmonics, which are integer multiples of the dominant frequency, play a significant role in shaping the waveform of a time series. By aligning the cutoff frequency with these harmonics, we keep relevant frequency components associated with the signal's structure and periodicity. This approach leverages the inherent relationship between frequencies to extract meaningful information while suppressing noise and irrelevant high-frequency components. The impact of COF on different harmonics' waveforms is shown in Fig. 3. We further elaborate on the impact of COF in our experimental results.

4 Experiments for Forecasting

4.1 Forecasting as Frequency Interpolation

Typically, the forecasting horizon is shorter than the given look-back window, rendering direct interpolation unsuitable. Instead, we formulate the forecasting task as the interpolation of a look-back window, with length L, to a combination of the look-back window and forecasting horizon, with length $L+H$. This design enables us to provide more supervision during training. With this approach, we can supervise not only the forecasting horizon but also the backcast task on the look-back window. Our experimental results demonstrate that this unique training strategy contributes to the improved performance of FITS. The interpolation rate of the forecasting task is calculated by:

$$\eta_{Fore} = 1 + \frac{H}{L},$$

where L represents the length of the look-back window and H represents the length of the forecasting horizon.

4.2 Experiment Settings

Datasets. All datasets used in our experiments are widely-used and publicly available real-world datasets, including, Traffic, Electricity, Weather, ETT (Zhou et al., 2021). We summarize the characteristics of these datasets in Tab. 1. Apart from these datasets for long-term time series forecasting, we also use the M4 dataset to test the short-term forecasting performance.

Table 1: The statistics of the seven used forecasting datasets.

Dataset	Traffic	Electricity	Weather	ETTh1&ETTh2	ETTm1 &ETTm2
Channels	862	321	21	7	7
Sampling Rate	1hour	1hour	10min	1hour	15min
Total Timesteps	17,544	26,304	52,696	17,420	69,680

Baselines. To evaluate the performance of FITS in comparison to state-of-the-art time series forecasting models, including PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023), FEDFormer (Zhou et al., 2022a), FiLM (Zhou et al., 2022b) and LTSF-Linear (Zeng et al., 2023), we directly refer to the reported results in the original papers under the same settings. We report the comparison with other transformer-based methods in the appendix.

Evaluation metrics. We follow the previous works (Zhou et al., 2022a; Zeng et al., 2022; Zhang et al., 2022) to compare forecasting performance using Mean Squared Error (MSE) as the core metrics. Moreover, to evaluate the short-term forecasting, we symmetric Mean Absolute Percentage Error (SMAPE) following TimesNet (Wu et al., 2023).

Implementation details. Following the settings of LTSF-Linear (Zeng et al., 2023), we set the look-back window of FITS as 720 for any forecasting horizon. Further experiments also show that a longer look-back window can result in better performance. To avoid information leakage, We choose the hyper-parameter based on the performance of the validation set.

4.3 Comparisons with SOTAs

Competitive Performance with High Efficiency

We present the results of our experiments on long-term forecasting in Tab. 2 and Tab. 3. The results for short-term forecasting on the M4 dataset are provided in the Appendix. Remarkably, our FITS consistently achieves comparable or even superior performance across all experiments.

Tab. 4 presents the number of trainable parameters for various TSF models using a look-back window of 96 and a forecasting horizon of 720 on the Electricity dataset. The table clearly demonstrates the exceptional efficiency of FITS compared to other models.

Among the listed models, the parameter counts range from millions down to thousands. Notably, large models such as TimesNet and Pyraformer require a staggering number of parameters, with

Table 2: Long-term forecasting results on ETT dataset in MSE. The best result is highlighted in **bold**, and the second best is highlighted with underline. IMP is the improvement between FITS and the second best/ best result, where a larger value indicates a better improvement.

Dataset	ETTh1				ETTh2				ETTm1				ETTm2			
Horizon	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
PatchTST	**0.370**	0.413	**0.422**	0.447	**0.274**	0.341	**0.329**	0.379	**0.293**	**0.333**	0.369	**0.416**	0.166	0.223	0.274	0.362
TimesNet	0.384	0.436	0.491	0.521	0.340	0.402	0.452	0.462	0.338	0.374	0.410	0.478	0.187	0.249	0.321	0.408
FEDFormer	0.376	0.420	0.459	0.506	0.346	0.429	0.496	0.463	0.379	0.426	0.445	0.543	0.203	0.269	0.325	0.421
FiLM	0.371	0.414	0.442	0.465	0.284	0.357	0.377	0.439	0.302	0.338	0.373	0.420	0.165	0.222	0.277	0.371
Dlinear	0.374	**0.405**	0.429	0.440	0.338	0.381	0.400	0.436	0.299	0.335	0.369	0.425	0.167	0.221	0.274	0.368
FITS	0.375	0.408	0.429	**0.427**	**0.274**	**0.333**	0.340	**0.374**	0.305	0.339	**0.367**	0.418	**0.164**	**0.217**	**0.269**	**0.347**
IMP	-0.005	-0.003	-0.007	0.013	0	0.008	-0.011	0.005	-0.012	-0.006	0.002	-0.002	0.002	0.004	0.005	0.015

Table 3: Long-term forecasting results on three popular datasets in MSE. The best result is highlighted in **bold** and the second best is highlighted with underline. IMP is the improvement between FITS and the second best/ best result, where a larger value indicates a better improvement.

Dataset	Electricity				Traffic				Weather			
Horizon	96	192	336	720	96	192	336	720	96	192	336	720
PatchTST	**0.129**	**0.147**	**0.163**	**0.197**	**0.360**	**0.379**	**0.392**	**0.432**	0.149	0.194	0.245	0.314
TimesNet	0.168	0.184	0.198	0.220	0.593	0.617	0.629	0.640	0.172	0.219	0.280	0.365
FEDFormer	0.193	0.201	0.214	0.246	0.587	0.604	0.621	0.626	0.217	0.276	0.339	0.403
FiLM	0.154	0.164	0.188	0.236	0.416	0.408	0.425	0.520	0.199	0.228	0.267	0.319
Dlinear	0.140	0.153	0.169	0.203	0.410	0.423	0.435	0.464	0.176	0.218	0.262	0.323
FITS	0.138	0.152	0.166	0.205	0.401	0.407	0.420	0.456	**0.145**	**0.188**	**0.236**	**0.308**
IMP	-0.009	-0.005	-0.003	-0.008	-0.041	-0.028	-0.028	-0.024	0.004	0.006	0.009	0.006

300.6M and 241.4M, respectively. Similarly, popular models like Transformer, Informer, Autoformer, and FEDformer have parameter counts in the range of 13.61M to 20.68M. Even the lightweight yet state-of-the-art model PatchTST has a parameter count of over 1 million.

In contrast, FITS stands out as a highly efficient model with an impressively low parameter count. With only 4.5K to 16K parameters, FITS achieves comparable or even superior performance compared to these larger models. It is worth highlighting that FITS requires significantly fewer parameters compared to the next smallest model, Dlinear, which has 139.7K parameters. For instance, when considering a 720 look-back window and a 720 forecasting horizon, the Dlinear model requires over 1 million parameters, whereas FITS achieves similar performance with only 10k-50k parameters.

Table 4: Number of trainable parameters and MACs of TSF models under look-back window=96 and forecasting horizon=720 on the Electricity dataset.

Model	Parameters	MACs
TimesNet	301.7M	1226.49G
Pyraformer	241.4M	0.80G
Transformer	13.61M	4.03G
Informer	14.38M	3.93G
Autoformer	14.91M	4.41G
FiLM	14.91M	5.97G
FEDformer	20.68M	4.41G
PatchTST	1.5M	5.07G
DLinear	139.7K	40M
FITS (Ours)	**4.5K$\sim$10K**	**1.6M$\sim$8.9M**

This analysis showcases the remarkable efficiency of FITS. Despite its small size, FITS consistently achieves competitive results, making it an attractive option for time series analysis tasks. FITS demonstrates that achieving state-of-the-art or close to state-of-the-art performance with a considerably reduced parameter footprint is possible, making it an ideal choice for resource-constrained environments.

Case Study on ETTh2 Dataset

We conduct a comprehensive case study on the performance of FITS using the ETTh2 dataset, which further highlights the impact of the look-back window and cutoff frequency on model performance. We provide a case study on other datasets in the Appendix. In our experiments, we observe that increasing the look-back window generally leads to improved performance, while the effect of increasing the cutoff frequency is minor.

Tab. 5 showcases the performance results obtained with different look-back window sizes and cutoff frequencies. Larger look-back windows tend to yield better performance across the board. On the other hand, increasing the cutoff frequency only results in marginal performance improvements. However, it is important to note that higher cutoff frequencies come at the expense of increased computational resources, as illustrated in Tab. 6.

Table 5: The results on the ETTh2 dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Look-back Window	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**2**	0.297687	0.296042	0.291606	0.289387	0.278644	0.278403	0.277708	0.27696
	3	0.297796	0.297377	0.290061	0.288239	0.277512	0.277746	0.276537	0.277068
	4	0.297106	0.295624	0.290725	0.287993	0.27624	0.27693	**0.274207**	0.274498
	5	0.296168	0.296698	0.288518	0.287375	0.276367	0.277935	0.275989	0.275636
192	**2**	0.380163	0.379868	0.360591	0.359769	0.336552	0.337976	0.334854	0.335887
	3	0.37983	0.381802	0.359088	0.359498	0.336384	0.336358	0.334666	0.335507
	4	0.379657	0.380439	0.359087	0.358536	0.334803	0.349995	0.333522	**0.333382**
	5	0.378556	0.379883	0.358809	0.359376	0.335451	0.343227	0.33384	0.335053
336	**2**	0.402706	0.404805	0.373257	0.374678	0.344241	0.344414	0.341869	0.342549
	3	0.403238	0.404878	0.372231	0.373948	0.345578	0.344976	0.341436	0.342793
	4	0.402702	0.407712	0.376199	0.374435	0.343004	0.344167	**0.340795**	0.342245
	5	0.403484	0.409516	0.375102	0.37462	0.344333	0.342731	0.341043	0.342214
720	**2**	0.420072	0.424272	0.403985	0.407392	0.379822	0.38519	0.376871	0.37677
	3	0.418323	0.420538	0.400986	0.40686	0.379638	0.386397	0.376236	0.376004
	4	0.417485	0.420982	0.399987	0.408128	0.379096	0.386409	0.375865	0.375637
	5	0.419122	0.420355	0.400776	0.407871	0.378665	0.390754	0.377138	**0.374586**

Considering these observations, we find utilizing a longer look-back window in combination with a low cutoff frequency to achieve near state-of-the-art performance with minimal computational cost. For instance, FITS surpasses other methods when employing a 720 look-back window and setting the cutoff frequency to the second harmonic. Remarkably, FITS achieves state-of-the-art performance with a parameter count of only around 10k. Moreover, by reducing the look-back window to 360, FITS already achieves close-to-state-of-the-art performance by setting the cutoff frequency to the second harmonic, resulting in a further reduction of the model's parameter count to under 5k (as shown in Tab. 6).

Table 6: The number of parameters under different settings on ETTh1 & ETTh2 dataset.

		Look-back Window			
Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**2**	703	1053	2279	5913
	3	1035	1820	4307	12064
	4	1431	2752	6975	20385
	5	1922	3876	10374	31042
192	**2**	1064	1431	2752	6643
	3	1564	2450	5192	13520
	4	2187	3698	8475	22815
	5	2914	5253	12558	34694
336	**2**	1615	1998	3483	7665
	3	2392	3395	6608	15704
	4	3321	5160	10725	26460
	5	4402	7293	15834	40006
720	**2**	3078	3510	5418	10512
	3	4554	5950	10266	21424
	4	6318	9030	16650	36180
	5	8370	12750	24570	54780

These results emphasize the lightweight nature of FITS, making it highly suitable for deployment and training on edge devices with limited computational resources. By carefully selecting the look-back window and cutoff frequency, FITS can achieve excellent performance while maintaining computational efficiency, making it an appealing choice for real-world applications.

5 Experiment for Anomaly Detection

5.1 Reconstruction as Frequency Interpolation

As discussed before, we tackle the anomaly detection tasks in the self-supervised reconstructing approach. Specifically, we make a N time down-sampling on the input and train a FITS network with an interpolation rate of $\eta_{Rec} = N$ to up-sample it.

5.2 Experiment Settings

Datasets. We use five commonly used benchmark datasets: SMD (Server Machine Dataset (Su et al., 2019)), PSM (Polled Server Metrics (Abdulaal et al., 2021)), SWaT (Secure Water Treatment (Mathur & Tippenhauer, 2016)), MSL (Mars Science Laboratory rover), and SMAP (Soil Moisture Active Passive satellite) (Hundman et al., 2018).

Baselines. We compare FITS with models such as TimesNet (Wu et al., 2023), Anomaly Transformer (Xu et al., 2022), THOC (Shen et al., 2020), Omnianomaly (Su et al., 2019). Following TimesNet (Wu et al., 2023), we also compare the anomaly detection performance with other models (Zeng et al., 2023; Zhang et al., 2022; Woo et al., 2022; Zhou et al., 2022a).

Evaluation metrics. Following the previous works (Xu et al., 2022; Shen et al., 2020; Wu et al., 2023), we use Precision, Recall, and F1-score as metrics.

Implementation details. We use a window size of 200 and downsample the time series segment by a factor of 4 to match the original segment during training with the FITS model. Anomaly detection follows the methodology of the Anomaly Transformer (Xu et al., 2022), where time points exceeding a certain reconstruction loss threshold are classified as anomalies. The threshold is selected based on the highest F1 score achieved on the validation set. To handle consecutive abnormal segments, we adopt a widely-used adjustment strategy (Su et al., 2019; Xu et al., 2018; Shen et al., 2020), considering all anomalies within a specific successive abnormal segment as correctly detected when one anomalous time point is identified. This approach aligns with real-world applications, where an abnormal time point often triggers the attention to the entire segment.

Table 7: Anomaly detection result of F1-scores on 5 datasets. The best result is highlighted in **bold**, and the second best is highlighted with underline. Full results are reported in the Appendix.

Models	FITS	TimesNet	Anomaly Transformer	THOC	Omni Anomaly	Stationary Transformer	LightTS	Dlinear	IMP
SMD	**99.95**	85.81	<u>92.33</u>	84.99	85.22	84.72	82.53	77.1	7.62
PSM	93.96	97.47	<u>97.89</u>	**98.54**	80.83	97.29	97.15	93.55	-3.93
SWaT	**98.9**	91.74	<u>94.07</u>	85.13	82.83	79.88	93.33	87.52	4.83
SMAP	70.74	71.52	**96.69**	<u>90.68</u>	86.92	71.09	69.21	69.26	-25.95
MSL	78.12	85.15	**93.59**	<u>89.69</u>	87.67	77.5	78.95	84.88	-15.47

5.3 Comparisons with SOTAs

As shown in Tab. 7, FITS achieves remarkable results on several datasets. Notably, on the SMD and SWaT datasets, FITS exhibits exceptional performance with F1-scores almost reaching perfection at around 99.95% and 98.9%, respectively. This demonstrates FITS' ability to accurately detect anomalies and classify them correctly. In comparison, other models, such as TimesNet, Anomaly Transformer, and Stationary Transformer, struggle to match FITS' performance on these datasets.

However, FITS shows comparatively lower performance on the SMAP and MSL datasets. These datasets present a challenge due to their binary event data nature, which may not be effectively captured by FITS' frequency domain representation. While models specifically designed for anomaly detection, such as THOC and Omni Anomaly, achieve higher F1-scores on these datasets.

For a more comprehensive evaluation, waveform visualizations and detailed analysis can be found in the appendix, providing deeper insights into FITS' strengths and limitations in different anomaly detection scenarios. It is important to note that the reported results are achieved with a parameter range of 1-4K and MACs (Multiply-Accumulate Operations) of 10-137K, which will be further detailed in the appendix.

6 Conclusions and Discussion

In this paper, we propose FITS for time series analysis, a low-cost model with $10k$ parameters that can achieve performance comparable to state-of-the-art models that are often several orders of magnitude larger. As a frequency-domain modeling technique, FITS has difficulty handling binary-valued time series and time series with missing data. For the former category, time-domain modeling is preferable as the raw data format is sufficiently compact. For the latter category, we could first employ simple yet effective time-domain imputation techniques and then apply FITS for efficient analysis.

References

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery; Data Mining*, KDD '21, pp. 2485–2494, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467174. URL `https://doi.org/10.1145/3447548.3467174`.

E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967. doi: 10.1109/MSPEC.1967.5217220.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, jul 2018. doi: 10.1145/3219819.3219845. URL `https://doi.org/10.11452F3219819.3219845`.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=cGDAkQo1C0p`.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms, 2022.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, 2022a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b. URL `https://openreview.net/forum?id=0EXmFzUn5I`.

Aditya P. Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ics security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pp. 31–36, 2016. doi: 10.1109/CySWater.2016.7469060.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. *CoRR*, abs/1905.10437, 2019. URL `http://arxiv.org/abs/1905.10437`.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13016–13026. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf`.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '19, pp. 2828–2837, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672. URL `https://doi.org/10.1145/3292500.3330672`.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting, 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*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Haowen Xu, Yang Feng, Jie Chen, Zhaogang Wang, Honglin Qiao, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, and Dan Pei. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. ACM Press, 2018. doi: 10.1145/3178876.3185996. URL `https://doi.org/10.1145/2F3178876.3185996`.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? 2023.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 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.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022a.

Tian Zhou, Ziqing Ma, xue wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL `https://openreview.net/forum?id=zTQdHSQUQWc`.

A Pipeline for Reconstruction

The pipeline for the reconstruction task is shown in Fig. 4. Note that the input is a downsampled time series segment, and the output is supervised on the original time series segment.

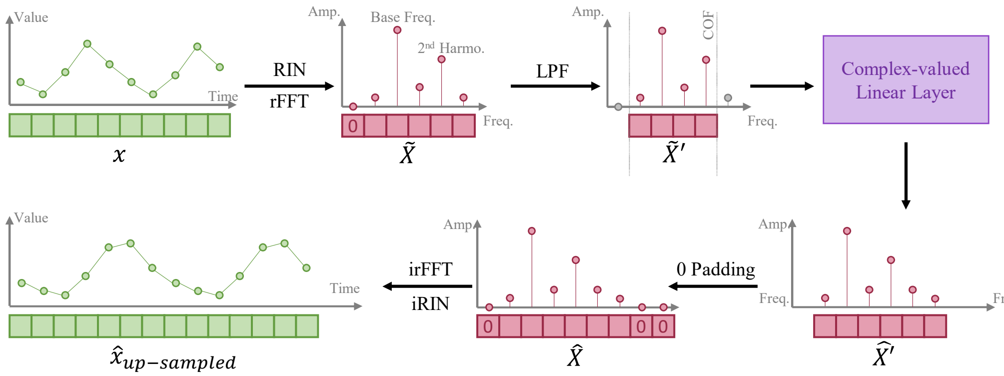


Figure 4: Pipeline of FITS, with a focus on the Reconstruction task.

B More Results on Forecasting Task

We show the comparison with transformer-based models, short-term forecasting on M4, and the impact of random seeds below.

B.1 Comparison with Transformer-based Methods

We further compare FITS with Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021) and Pyraformer (Liu et al., 2022b). The results are shown in Tab. 8 and Tab. 9.

Table 8: Long-term forecasting results on ETT datasets in MSE. The best result is highlighted in **bold**.

Dataset	ETTh1				ETTh2				ETTm1				ETTm2			
Horizon	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
Autoformer	0.449	0.500	0.521	0.514	0.358	0.456	0.482	0.515	0.505	0.553	0.621	0.671	0.255	0.281	0.339	0.433
Informer	0.865	1.008	1.107	1.181	3.755	5.602	4.721	3.647	0.672	0.795	1.212	1.166	0.365	0.533	1.363	3.379
FEDFormer	0.376	0.420	0.459	0.506	0.346	0.429	0.496	0.463	0.379	0.426	0.445	0.543	0.203	0.269	0.325	0.421
Pyraformer	0.664	0.790	0.891	0.963	0.645	0.788	0.907	0.963	0.543	0.557	0.754	0.908	0.435	0.730	1.201	3.625
FITS	**0.375**	**0.408**	**0.429**	**0.427**	**0.274**	**0.333**	**0.340**	**0.374**	**0.305**	**0.339**	**0.367**	**0.418**	**0.164**	**0.217**	**0.269**	**0.347**

Table 9: Long-term forecasting results on three popular datasets in MSE. The best result is highlighted in **bold**.

Dataset	Electricity				Traffic				Weather			
Horizon	96	192	336	720	96	192	336	720	96	192	336	720
Autoformer	0.201	0.222	0.231	0.254	0.613	0.616	0.622	0.660	0.266	0.307	0.359	0.419
Informer	0.274	0.296	0.300	0.373	0.719	0.696	0.777	0.864	0.300	0.598	0.578	1.059
FEDFormer	0.193	0.201	0.214	0.246	0.587	0.604	0.621	0.626	0.217	0.276	0.339	0.403
Pyraformer	0.386	0.386	0.378	0.376	2.085	0.867	0.869	0.881	0.896	0.622	0.739	1.004
FITS	**0.138**	**0.152**	**0.166**	**0.205**	**0.401**	**0.407**	**0.420**	**0.456**	**0.145**	**0.188**	**0.236**	**0.308**

B.2 Short-term Forecasting on M4

We evaluate FITS' performance on the M4 dataset following the TimesNet (Wu et al., 2023). We retrieve the following results from the TimesNet paper. As shown in Tab.10, FITS shows the suboptimal results on the M4 dataset. The reason for this outcome is threefold. First, the M4 dataset is a collection of many time series from different domains. These time series have different temporal information and periodicity, and no correlations exist among them. We can not regard them as simple multivariate forecasting tasks. Second, other models have a very large amount of parameters,

Table 10: Results on M4 dataset in SMAPE.

	FITS	DLinear	TimesNet	N-Hits	N-Beats
Yearly	14.00	16.96	13.38	13.41	13.43
Quarterly	10.72	12.14	10.1	10.2	10.12
Monthly	13.49	13.51	12.67	12.7	12.67

especially TimesNet, which makes them have enough capability to model such diverse datasets with one model. However, considering the lightweight of FITS, it is hard for it to achieve ideal results. Finally, the setting for the M4 dataset is not suitable for FITS. The look-back window is set to 12, 16, and 36 for yearly, quarterly, and monthly prediction accordingly, which is twice the length of the forecasting horizon. Such a short look-back window is very difficult to extract meaningful frequency representation, which further worsens the FITS' performance. We compare FITS with lightweight model DLinear (Zeng et al., 2022), state-of-the-art model TimesNet (Wu et al., 2023) and two hierarchical time series modeling model N-Hits (Challu et al., 2022) and N-Beats (Oreshkin et al., 2019).

B.3 Impact of Random Seeds

We run the experiment 4 times with different chosen random seeds (i.e., 114, 514, 1919, 810) to get the standard deviation. As shown in Tab. 11, random seeds make very little impact on the FITS. Thanks to the small number of parameters, FITS is very robust to such noise.

Table 11: Table of mean and standard deviation (std) of FITS on ETTh2 dataset. The data is in mean(std) format.

Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**2**	0.295414(1.03e-07)	0.290974(3.18e-07)	0.281156(1.0334e-05)	0.276398(1.35e-07)
	3	0.294779(1.5e-07)	0.297235(1.73e-04)	0.284811(1.175e-04)	0.27579(2.088e-06)
	4	0.29443(6.7e-08)	0.289059(8.5e-08)	0.27689(9.2e-08)	0.2733(1.5e-08)
	5	0.294409(1.194e-06)	0.291449(1.5622e-05)	0.277162(1.87e-07)	0.273262(2.9e-08)
192	**2**	0.378869(2.763e-06)	0.361255(5.69e-07)	0.337159(9.6e-08)	0.334243(1e-09)
	3	0.377842(8.3e-08)	0.360074(2.597e-06)	0.336451(3.1e-08)	0.333356(4e-09)
	4	0.377441(3.04e-07)	0.359773(4.75e-07)	0.335045(9e-09)	0.332138(3e-09)
	5	0.377859(7.4e-08)	0.358795(2.5e-07)	0.337158(1.2363e-05)	0.332015(1.8e-08)
336	**2**	0.401767(8.53e-07)	0.37311(2.39e-07)	0.343065(2.3e-08)	0.341445(2.849e-06)
	3	0.402027(6.81e-07)	0.37219(5.6e-07)	0.342487(7e-09)	0.340162(2e-09)
	4	0.403194(7.561e-06)	0.373196(8.08e-07)	0.341473(7e-09)	0.33921(5e-09)
	5	0.404614(4.802e-05)	0.372623(5.05e-07)	0.341668(5.9e-08)	0.339122(1.2e-08)
720	**2**	0.416372(6.678e-06)	0.400324(6.35e-07)	0.380281(4.9e-08)	0.374877(1.09e-07)
	3	0.414283(2.18e-07)	0.410352(4.126e-04)	0.379871(1.39e-07)	0.374562(6e-09)
	4	0.415606(2.705e-06)	0.398548(2.17e-07)	0.378965(5.4e-08)	0.373264(2.8e-08)
	5	0.414254(4.21e-07)	0.401551(2.6431e-05)	0.378391(9e-09)	0.373222(3e-09)

C Case Study on Other Datasets

We show the parameter table and performance on other datasets below.

C.1 ETTm1 & m2

Tab. 12 shows the parameter count of parameters of FITS with different settings on the ETTm1 & 2 datasets. Tab. 13 and Tab.14 show the corresponding results on ETTm1 and ETTm2 datasets with different settings. Note that FITS constantly achieves SOTA performance on the ETTm2 dataset with under 10k parameters.

Table 12: The number of parameters under different settings on ETTm1 & ETTm2 dataset.

		Look-back Window			
Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**4**	420	513	621	1330
	6	561	759	1015	2444
	8	703	1053	1505	3835
	10	861	1426	2050	5609
	12	1035	1820	2726	7636
192	**4**	645	703	759	1505
	6	850	1035	1218	2726
	8	1064	1431	1820	4307
	10	1302	1922	2501	6248
	12	1564	2450	3290	8549
336	**4**	990	969	966	1715
	6	1275	1449	1566	3149
	8	1615	1998	2275	5015
	10	1974	2666	3157	7242
	12	2392	3395	4136	9960
720	**4**	1890	1710	1518	2380
	6	2448	2530	2436	4324
	8	3078	3510	3570	6844
	10	3780	4650	4920	9940
	12	4554	5950	6486	13612

Table 13: The results on the ETTm1 dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Input length	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**4**	0.36539	0.368287	0.315333	0.314558	0.311276	0.310786	0.323238	0.321785
	6	0.366902	0.36644	0.316294	0.314923	0.308139	0.307256	0.320556	0.318702
	8	0.363857	0.364832	0.314383	0.314648	**0.305866**	0.306163	0.313537	0.315031
	10	0.365007	0.366312	0.313453	0.312554	0.30812	0.308093	0.313483	0.315461
	12	0.362493	0.364372	0.314225	0.314401	0.306075	0.306781	0.313896	0.316464
192	**4**	0.402017	0.411609	0.350637	0.351265	0.344014	0.342416	0.347725	0.3501
	6	0.402813	0.413139	0.349372	0.352178	0.339382	0.34013	0.344214	0.345845
	8	0.403378	0.408561	0.350732	0.35014	0.340666	0.339582	0.341009	0.341524
	10	0.402122	0.409548	0.35038	0.351084	0.340434	0.339451	0.341734	0.343237
	12	0.404392	0.416905	0.348248	0.350706	0.339771	**0.339245**	0.342307	0.341273
336	**4**	0.457025	0.626306	0.38627	0.498018	0.376918	0.378298	0.373669	0.375122
	6	0.453296	0.604673	0.387365	0.444529	0.37518	0.374951	0.372264	0.371787
	8	0.478034	0.631451	0.387241	0.440868	0.373784	0.374112	0.368906	0.37
	10	0.5171	0.629327	0.394267	0.468663	0.374651	0.373348	**0.367833**	0.369997
	12	0.443713	0.63693	0.386978	0.412943	0.373993	0.374413	0.370057	0.368717
720	**4**	0.593805	0.664672	0.460092	0.634543	0.431741	0.459255	0.422784	0.422793
	6	0.599244	0.677897	0.47532	0.665818	0.430778	0.456598	0.422234	0.423324
	8	0.620804	0.70239	0.462694	0.620843	0.433994	0.474499	**0.418575**	0.42085
	10	0.564101	0.723161	0.459982	0.653963	0.432989	0.470532	0.418746	0.419788
	12	0.604411	0.730127	0.542039	0.625938	0.432008	0.485034	0.420789	0.424112

C.2 Traffic

Tab. 15 shows the parameter count of parameters of FITS with different settings on the Traffic dataset. Tab. 16shows the result on the Traffic dataset with different settings correspondingly. The traffic dataset has a very large amount of channels, making many models need many parameters to model the temporal information. FITS only needs 50k parameters to achieve comparable performance.

Table 14: The results on the ETTm2 dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Input length	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**4**	0.189949	0.187104	0.175861	0.175653	0.168331	0.167802	0.167627	0.168365
	6	0.187921	0.187752	0.175081	0.174664	0.167421	0.166934	0.165331	0.166699
	8	0.18755	0.186862	0.174284	0.174376	0.167456	0.166398	0.16545	0.165945
	10	0.186856	0.187068	0.174272	0.174055	0.166025	0.166027	0.164797	0.165304
	12	0.188115	0.187032	0.174164	0.17395	0.166229	0.16578	**0.16419**	0.164371
192	**4**	0.250291	0.250258	0.235167	0.235682	0.222561	0.221472	0.221164	0.2204
	6	0.25162	0.251188	0.234177	0.234117	0.222139	0.221428	0.219001	0.218901
	8	0.250965	0.252477	0.234083	0.234356	0.221141	0.221143	0.219023	0.21849
	10	0.251273	0.252961	0.233744	0.23406	0.220952	0.220169	0.218367	0.217286
	12	0.250632	0.250457	0.233587	0.234173	0.22108	0.220789	0.217687	**0.217022**
336	**4**	0.311742	0.314966	0.289996	0.28909	0.27523	0.275501	0.272427	0.271816
	6	0.311689	0.315261	0.289143	0.289707	0.275103	0.275547	0.271085	0.270759
	8	0.317793	0.319121	0.288993	0.294166	0.274604	0.274989	0.270647	0.270937
	10	0.311076	0.326517	0.289358	0.290076	0.274078	0.274672	0.270995	0.270266
	12	0.311036	0.323915	0.288891	0.291164	0.273783	0.274388	0.269596	**0.269525**
720	**4**	0.41408	0.420778	0.385617	0.407686	0.365705	0.367553	0.350079	0.349886
	6	0.412397	0.423905	0.385204	0.410507	0.36524	0.369753	0.349508	0.348787
	8	0.418551	0.43163	0.386254	0.406818	0.365354	0.371821	0.349908	0.349498
	10	0.415603	0.427358	0.387376	0.411223	0.364901	0.371391	0.348837	0.347984
	12	0.420396	0.43113	0.394693	0.404091	0.365673	0.370805	0.348593	**0.347862**

Table 15: The number of parameters under different settings on Traffic dataset.

		Look-back Window			
Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**3**	1035	1820	4307	12064
	4	1431	2752	6975	20385
	5	1922	3876	10374	31042
192	**3**	1564	2450	5192	13520
	4	2187	3698	8475	22815
	5	2914	5253	12558	34694
336	**3**	2392	3395	6608	15704
	4	3321	5160	10725	26460
	5	4402	7293	15834	40006
720	**3**	4554	5950	10266	21424
	4	6318	9030	16650	36180
	5	8370	12750	24570	54780

C.3 Weather

Tab. 17 shows the parameter count of parameters of FITS with different settings on the Weather dataset. Tab. 16shows the result on the Traffic dataset with different settings correspondingly. Note that we achieve the result in the main table by setting the COF as 75 and the look-back window as 700.

Table 16: The results on the Traffic dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Input length	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**3**	0.694065	0.694425	0.474606	0.475881	0.455815	0.457292	0.436317	0.436616
	4	0.690741	0.691064	0.462886	0.463642	0.434575	0.434842	0.414185	0.415293
	5	0.688774	0.691499	0.459929	0.459652	0.423814	0.422843	**0.401225**	0.403405
192	**3**	0.627212	0.636434	0.481686	0.485085	0.463516	0.46417	0.442661	0.443547
	4	0.625307	0.649024	0.470148	0.483849	0.439995	0.440732	0.4198	0.419938
	5	0.623088	0.636091	0.466362	0.478839	0.429684	0.4296	**0.407131**	0.408353
336	**3**	0.635301	0.662283	0.4962	0.510793	0.47309	0.476491	0.454243	0.456989
	4	0.63295	0.656833	0.484066	0.50553	0.451054	0.454847	0.432025	0.433721
	5	0.631095	0.670716	0.480058	0.512274	0.439686	0.444552	**0.420825**	0.423244
720	**3**	0.685472	0.732168	0.529004	0.606921	0.500635	0.587891	0.488116	0.489934
	4	0.684401	0.752384	0.515154	0.617825	0.481284	0.58569	0.468166	0.469335
	5	0.688761	0.752565	0.523269	0.628914	0.472644	0.583838	**0.456807**	0.460696

Table 17: The number of parameters under different settings on Weather dataset.

		Look-back Window			
Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**3**	364	408	480	899
	4	420	513	621	1330
	5	496	630	806	1845
	8	703	1053	1505	3835
192	**3**	560	561	580	1015
	4	645	703	759	1505
	5	752	861	988	2050
	8	1064	1431	1820	4307
336	**3**	854	765	720	1189
	4	990	969	966	1715
	5	1136	1197	1248	2378
	8	1615	1998	2275	5015
720	**3**	1638	1360	1140	1624
	4	1890	1710	1518	2380
	5	2160	2100	1950	3280
	8	3078	3510	3570	6844

C.4 Electricity

Tab. 19 shows the parameter count of parameters of FITS with different settings on the Electricity dataset. Tab. 20 shows the result on the Electricity dataset with different settings correspondingly. We find that the Electricity dataset is sensitive to the COF. This is because this dataset shows significant

Table 18: The results on the Weather dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Input length	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**3**	0.197956	0.198834	0.190438	0.190583	0.177569	0.176701	0.174107	0.173947
	4	0.19808	0.198548	0.190979	0.19016	0.176951	0.175991	0.172446	0.172651
	5	0.198305	0.197615	0.189992	0.190143	0.175894	0.176468	0.172261	0.173201
	8	0.197515	0.197714	0.189467	0.190344	0.175324	0.174741	0.171744	**0.170606**
192	**3**	0.243689	0.244304	0.234231	0.233943	0.219906	0.219789	0.216264	0.21634
	4	0.243442	0.244047	0.233548	0.233765	0.219619	0.2193	0.215846	0.215159
	5	0.244325	0.244027	0.232503	0.233373	0.218952	0.219246	0.215042	0.214364
	8	0.243439	0.244354	0.233635	0.232369	0.218155	0.218985	0.214377	**0.214347**
336	**3**	0.296318	0.299125	0.281715	0.284938	0.266293	0.266286	0.260424	0.259934
	4	0.295563	0.298132	0.281794	0.284505	0.266475	0.266438	0.260124	0.259734
	5	0.295225	0.299156	0.281916	0.287063	0.265812	0.26592	**0.259221**	0.259553
	8	0.295462	0.301229	0.281217	0.288032	0.26527	0.265357	0.259368	0.259352
720	**3**	0.368714	0.373197	0.353147	0.358274	0.333046	0.334346	0.321251	0.32157
	4	0.369893	0.374172	0.352436	0.355783	0.332602	0.335239	0.32068	0.322293
	5	0.3691	0.373669	0.352941	0.358575	0.332862	0.336928	0.321146	0.321193
	8	0.368921	0.376838	0.353168	0.361575	0.33276	0.334887	**0.32057**	0.321736

multi-periodicity, which requires capturing high-frequency components. Otherwise, FITS will not learn such information.

Table 19: The number of parameters under different settings on Electricity dataset.

		Look-back Window			
Horizon	COF/nth Harmonic	**90**	**180**	**360**	**720**
96	**2**	703	1053	2279	5913
	3	1035	1820	4307	12064
	4	1431	2752	6975	20385
	5	1922	3876	10374	31042
	8	3698	8475	24186	75628
192	**2**	703	1053	2279	5913
	3	1035	1820	4307	12064
	4	1431	2752	6975	20385
	5	1922	3876	10374	31042
	8	3698	8475	24186	75628
336	**2**	1615	1998	3483	7665
	3	2392	3395	6608	15704
	4	3321	5160	10725	26460
	5	4402	7293	15834	40006
	8	8514	15900	36974	97902
720	**2**	3078	3510	5418	10512
	3	4554	5950	10266	21424
	4	6318	9030	16650	36180
	5	8370	12750	24570	54780
	8	16254	27750	57546	133644

Table 20: The results on the Electricity dataset. Values are visualized with a green background, where darker background indicates worse performance. The top-5 best results are highlighted with a red background, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

	Input length	90		180		360		720	
Horizon	COF/nth Harmonic	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**	**F**	**B+F**
96	**2**	0.219861	0.220638	0.180801	0.181306	0.187296	0.18764	0.182412	0.182459
	3	0.214897	0.220059	0.170256	0.170923	0.167465	0.167297	0.162607	0.16269
	4	0.211179	0.211584	0.164911	0.164888	0.156822	0.156631	0.151266	0.15207
	5	0.20964	0.210331	0.1614	0.162626	0.151409	0.15133	0.1457	0.146501
	8	0.205661	0.207206	0.155651	0.156382	0.142126	0.142532	0.139022	**0.13841**
192	**2**	0.216865	0.22451	0.195207	0.192494	0.200281	0.20018	0.195381	0.195814
	3	0.211918	0.224618	0.191438	0.182895	0.180623	0.180136	0.175342	0.175513
	4	0.210492	0.223837	0.179412	0.18544	0.169962	0.169827	0.164689	0.165129
	5	0.207801	0.217388	0.17504	0.177671	0.164243	0.165165	0.159636	0.160017
	8	0.205524	0.216681	0.169276	0.194963	0.155918	0.157119	0.154376	**0.152842**
336	**2**	0.232701	0.248004	0.213782	0.236475	0.2144	0.215592	0.209039	0.20925
	3	0.228432	0.253425	0.205334	0.230306	0.196429	0.195466	0.189955	0.189872
	4	0.223753	0.243039	0.207252	0.225273	0.18502	0.185789	0.180464	0.17935
	5	0.22162	0.252762	0.195054	0.23265	0.184309	0.181705	0.174788	0.175183
	8	0.220267	0.247537	0.190028	0.242119	0.171824	0.175581	0.166873	**0.166276**
720	**2**	0.279359	0.304686	0.25978	0.311533	0.255001	0.285431	0.249335	0.249288
	3	0.271019	0.303177	0.250443	0.319552	0.234653	0.266245	0.230563	0.23081
	4	0.272873	0.30215	0.267512	0.348516	0.228953	0.266807	0.219166	0.224155
	5	0.274384	0.308015	0.283174	0.346803	0.222556	0.259166	0.212817	0.215066
	8	0.274878	0.320414	0.271509	0.383154	0.221972	0.298277	**0.205646**	0.210626

D Full Anomaly Detection Results

The full results with Accuracy, Precision, Recall, and F1-score are shown in Tab. 21. For better performance, we also conduct experiments only on the first channel of the SML dataset, denoted as (C0). We also trained FITS using only the analog channels of SWaT, denoted as (analog).

Table 21: Full results on five datasets.

Datasets	Accuracy	Precision	Recall	F1-score
SMD	99.92	99.9	100	99.95
PSM	94.43	97.2	90.43	93.69
SWaT	99.42	97.84	100	98.9
SWaT(analog)	97.81	91.74	100	95.69
SMAP	89.39	77.52	65.05	70.74
MSL	81.52	61.38	80.16	69.52
MSL(C0)	83.77	81.34	75.15	78.12

E Datasets Visualization on Anomaly Detection

As shown in Fig. 5 and Fig. 6, most PSM and SMD datasets channels are analog values. Especially the PSM dataset shows great periodicity.

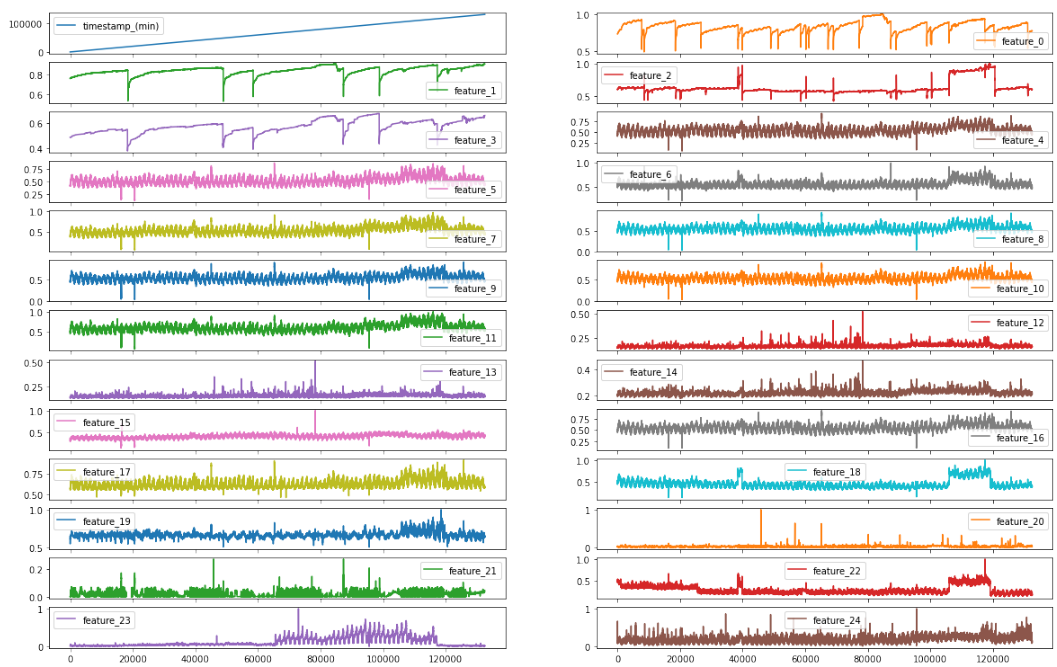


Figure 5: Waveform of PSM dataset.

While some channels in the SWaT dataset are binary event values, as shown in Fig. 7.

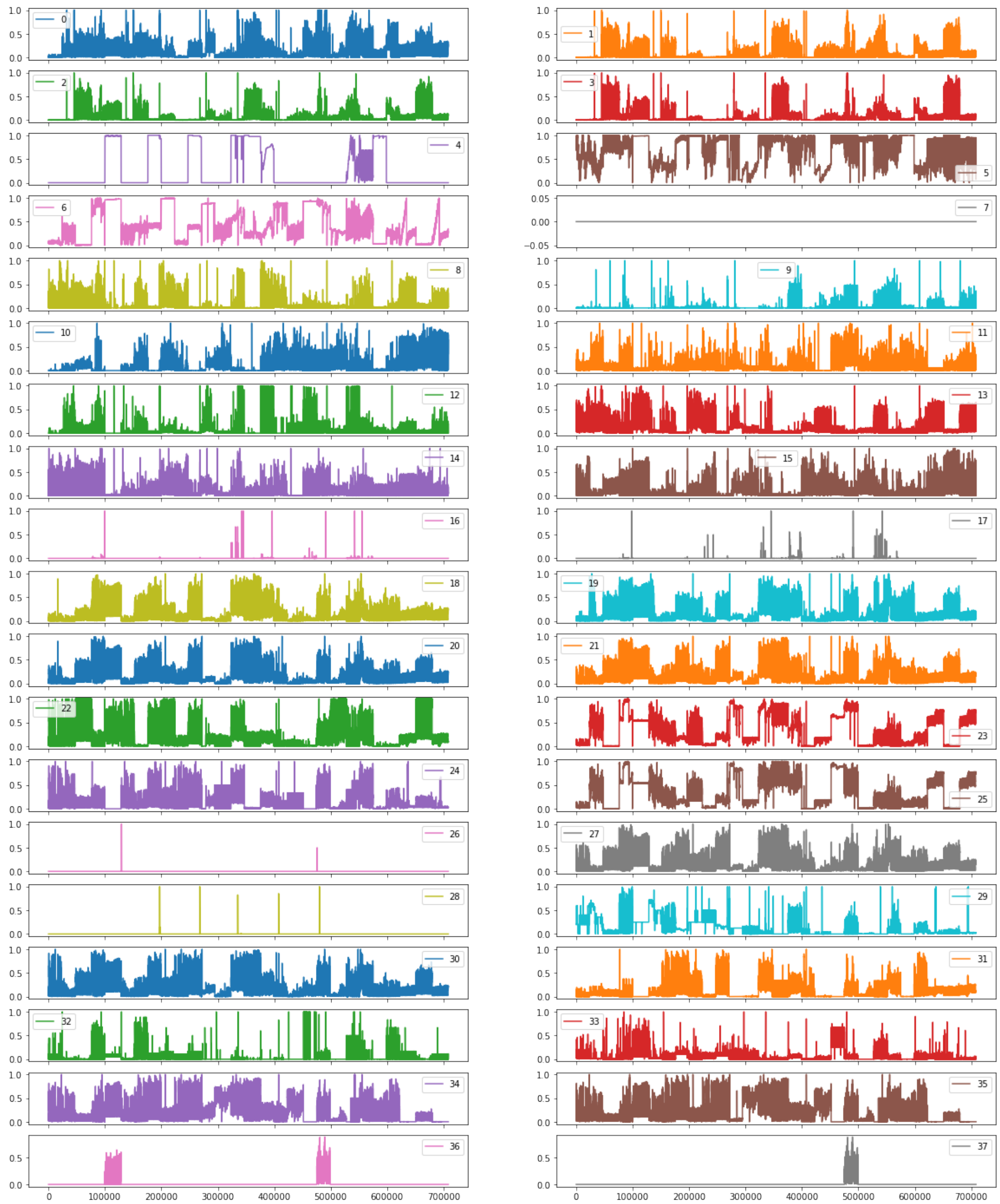


Figure 6: Waveform of SMD dataset.

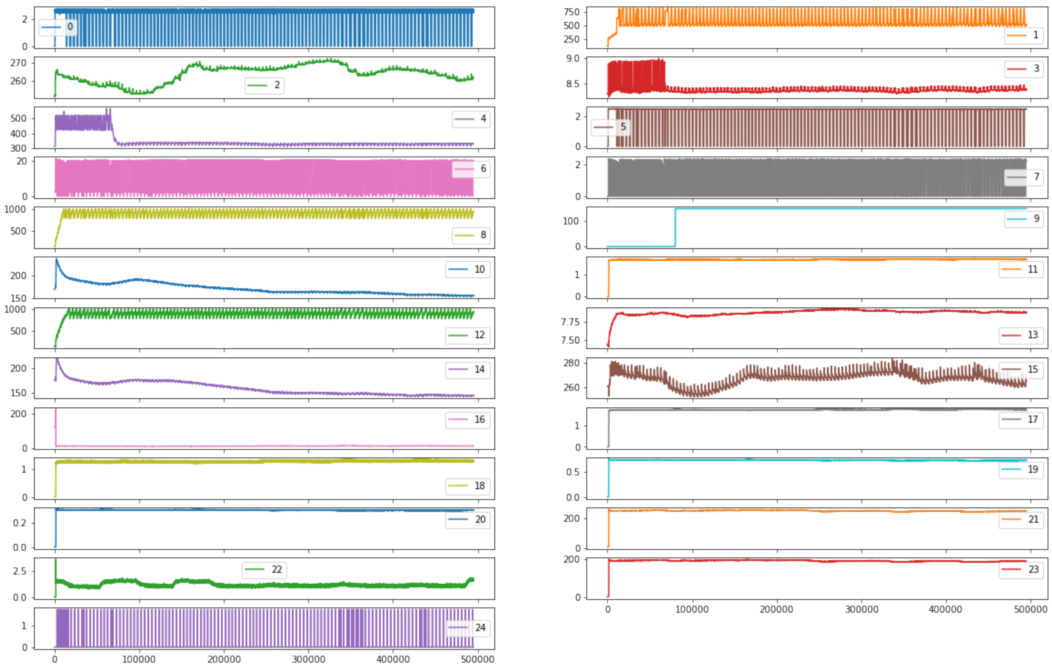


Figure 7: Waveform of SWAT dataset.

However, as shown in Fig. 8 and Fig. 9, for SMAP and MSL datasets, most channels are binary event values that are hard for FITS to learn frequency representation.

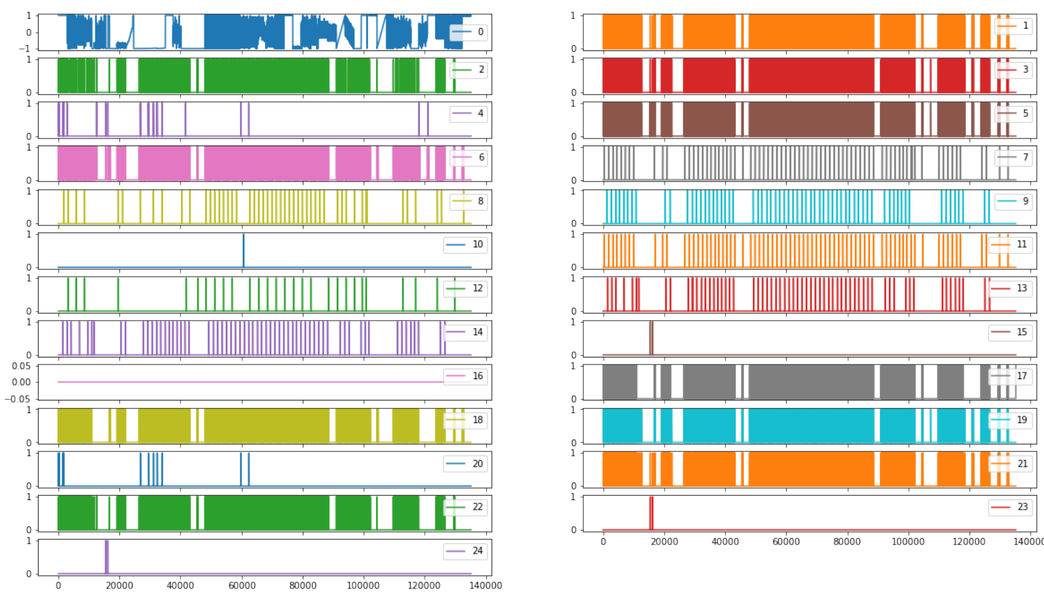


Figure 8: Waveform of SMAP dataset.

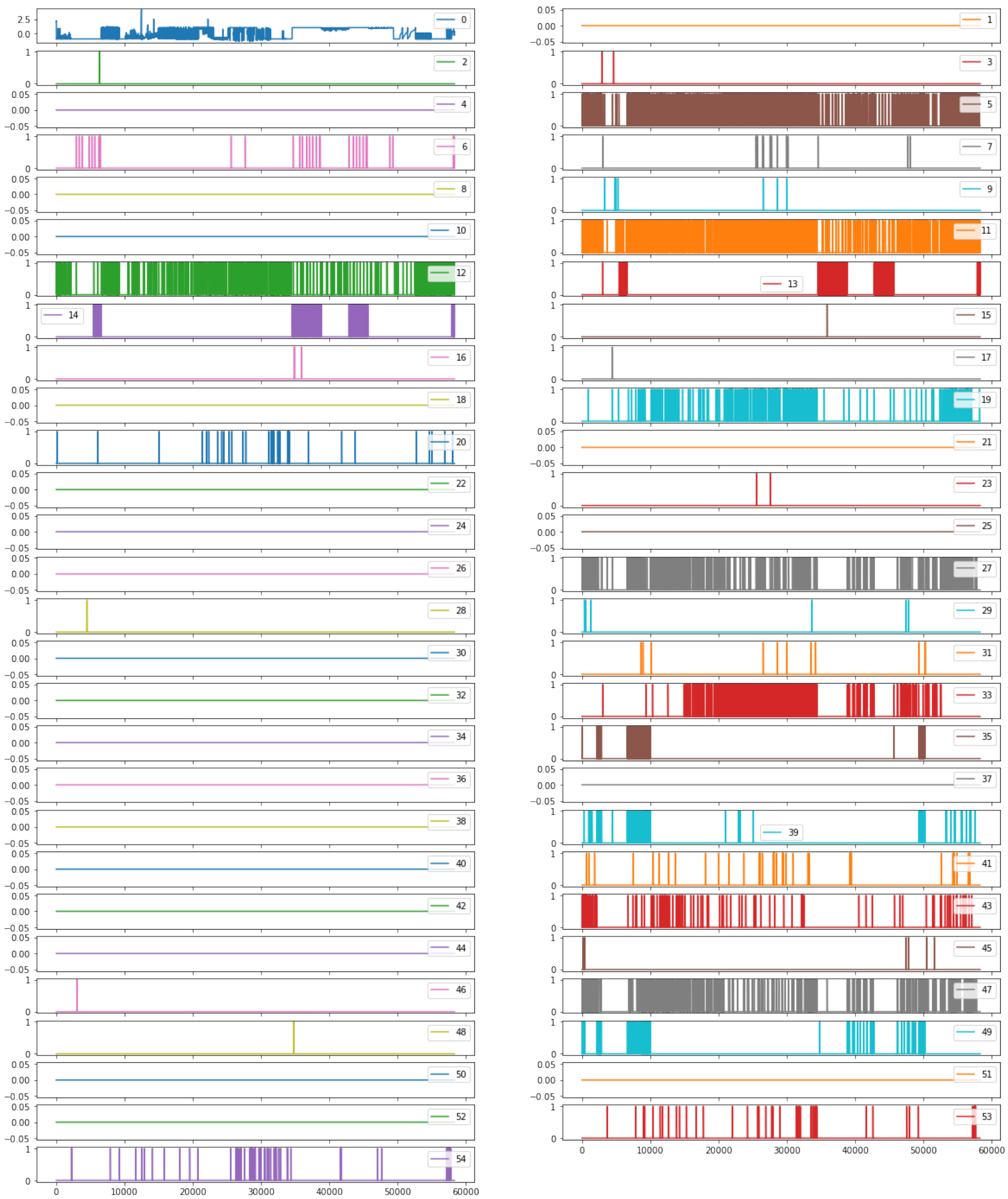


Figure 9: Waveform of MSL dataset.

F Parameter Counts for Anomaly Detection

We use a fixed sliding window of 200 and 400 for all the datasets and do not apply any frequency filter. The downsample rate is set as 4 for any dataset. Thus, the number of parameters is as Tab. 22.

Table 22: MACs and parameter count of FITS on Anomaly Detection task. We report the MACs on the SWaT dataset which has 55 channels.

Window	Params	MACs
200	2600	137.5k
400	10200	550k