TIMEPATCH: DIFFUSION ON NON-STATIONARY AND NOISY TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of Transformer-based systems in vision and language-related tasks led to the development of state-of-the-art approaches in time series. However, there are several fundamental problems with existing methods and their benchmarks. Although the real-world data is non-stationary, existing datasets are mostly presented with stationary data. This entails models' focus shift towards learning simple patterns. As a result, models struggle with predicting noisy data, such as stock data and Brownian Motion. In this work we present TimePatch Diffusion, which is a novel architecture, to deal with complex time-series forecasting tasks. It is composed of a DDPM encoder and a TPatch decoder. The former learns time-series noise, while the latter extracts encoded features from captured representations.

1 INTRODUCTION

Time Series Forecasting (TSF) is a prevalent problem with applications in various fields, such as healthcare, finance, traffic, and weather. With the advancement of Deep Neural Networks (DNNs), deep learning models have significantly surpassed classical statistical models. In particular, Transformer-based models (Vaswani et al., 2017b; Nie et al., 2023) have created a remarkable performance leap in TSF problems.

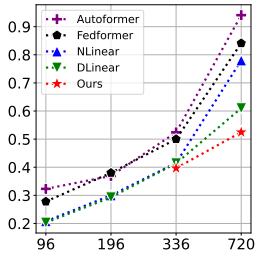


Figure 1: **MAE on Exchange Rate.** Existing models decrease exponentially in performance while ours linearly decrease.

The core of the Transformer-based model lies in its Self-Attention mechanism (Vaswani et al., 2017b), which enables the capture of inter-data correlations. However, it has two fundamental limitations for time-series forecasting. The first is permutation invariance, which results in its inability to capture sequential relationships. Most existing models address this by applying different positional embedding techniques. The second limitation is the computational complexity of the Self-Attention module. With a complexity of $\mathcal{O}(n^2)$, where n is the sequence length, it becomes challenging for the model to capture long-term relationships. Some existing approaches Informer (Zhou et al., 2020), Longformer (Beltagy et al., 2020) decrease the number of operations by using sparsity, while others Fedformer (Zhou et al., 2022), Autoformer (Wu et al., 2021) utilize the domain change approach.

To address such issues, existing TSF models focus on the fundamental components of the time series data, such as trend and seasonality. By decomposing the time series into trend and remainder components, models can achieve state-of-the-art performance. However, these properties are only distinguishable when the data exhibits certain level of stationarity. As noted in Zeng et al. (2022), in stationary cases, even simple linear models can produce reasonable results. We found that the majority of the widely used TSF benchmark datasets are stationary, except for the illness and exchange rate datasets. This observation highlights a direct relationship between the success of existing TSF models that utilize decomposition and the stationarity property of the data. However, financial data, which is highly uncorrelated, is an exception. State-of-the-art TSF models exhibit significant performance drops in predicting long-term exchange rate data, indicating that the exchange rate dataset lacks obvious trend and seasonality patterns (Liu et al., 2022).

To overcome the limitations of attention-based deterministic models when dealing with non-stationary time series data, we propose a non-deterministic model with uncertainty. Several generative models can operate with noisy data, including denoising autoencoders (Vincent et al., 2008), generative adversarial networks (Goodfellow et al., 2020), and diffusion models (Ho et al., 2020b). However, the former models often suffer from low fidelity, non-convergence, and mode collapse. In contrast, diffusion models mitigate these problems by utilizing slow noising and denoising processes. Specifically, the forward chain applies noise to the original data until it becomes white noise, while the reverse process reconstructs the data from the noise using a neural network (Sohl-Dickstein et al., 2015). Such a system is highly beneficial in modeling noisy uncorrelated data.

In this paper, we propose a novel approach named *TPatch-Diffusion* to address the issue of non-stationary chaotic TSF. The model consists of two primary components: a Denoising Diffusion Probabilistic Model (*DDPM*) Ho et al. (2020b) operating as an encoder and a patching module (*TPatch*) serving as a new decoder. The diffusion module is designed to introduce uncertainty to the model and learn the noise inherent representations in the time series data. On the other hand, *NTPatch* extracts the structure of the encoded features obtained from diffusion.

Therefore, our contributions are as follows,

- We examine the common TSF benchmark dataset from the standpoint of stationarity and its impact on existing TSF models.
- We propose a novel TSF method that combines DDPM with a new decoder named TPatch.
- We demonstrate that TPatch-Diffusion outperforms existing models on market datasets.

2 BACKGROUND

2.1 STATIONARITY OF EXISTING TSF METHODS

TSF models that use decomposition divide time series into trend and seasonality, utilizing both pieces of information while assuming the stationarity of the dataset. Since such existing TSF models utilize the concept of series decomposition, the data is divided into trend and seasonality, which enables the learning of patterns in either of the components. The ability to capture meaningful representations of the trend and seasonality makes models possible to precisely forecast future values.

Decomposition extracts information on seasonality by smoothing original data with trends. As a result, the model performs effectively when there is a substantial amount of meaningful information extracted from the seasonality component. As stationarity is an essential property of a time series as assumed in the existing TSF model, we aim to analyze the effect of stationarity on the existing TSF models. To examine the stationarity of the current TSF benchmark datasets, we conducted simple Augmented Dickey–Fuller (ADF) (Dickey & Fuller, 1979) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) (Kwiatkowski et al., 1992) tests.

Dataset	ADF p-value	KPSS p-value	Conclusion	PatchTST MSE Loss
ETTh1	0.2456	0.0100	Trend Stationary	0.447
ETTh2	0.1958	0.0100	Trend Stationary	0.379
Electricity	$4.8925 \cdot 10^{-17}$	0.0100	Trend Stationary	0.197
Weather	0.0000	0.1000	Stationary	0.314
Traffic	$7.4316 \cdot 10^{-23}$	0.0100	Trend Stationary	0.432
Illness	0.2717	0.0100	**Non-Stationary**	1.470
Exchange Rate	0.4331	0.0100	**Non-Stationary**	0.854
QQQ Stock	0.2975	0.0100	Trend-Stationary	0.250
SPY Stock	0.0550	0.0100	**Non-Stationary**	1.363
MSFT Stock	0.2380	0.0100	Trend-Stationary	0.662
NFLX Stock	0.7799	0.0100	**Non-Stationary**	1.363

Table 1: Datasets Stationarity Test Results

Table 1 demonstrates that the majority of the data exhibits stationary characteristics, which is a significant factor in the success of the existing model with the widely used TSF benchmark dataset. The values, combined with the results of the tests and the loss values of the PatchTST (Nie et al., 2023), clearly indicate that model performance has a direct correlation with the stationarity property of the data.

However, upon analyzing the performance of the current TSF model on the non-stationary data in the dataset, i.e., the exchange rate, it becomes evident that the model's performance exponentially declines as the prediction horizon increases. Non-stationary data lacks obvious patterns for the model to learn, resulting in a drop in performance. In an effort to stationarize the data, Liu et al. (2022) suggests using a local stationarization module, which further enhances performance on stationary datasets but does not improve performance on fully non-stationary data.

Additionally, *NLinear* (Zeng et al., 2022), demonstrates lower loss values than repeating the most last values, which prompts further analysis of the forecasted values. NLinear incorporates the momentum characteristic, which utilizes the most recent value and leverages the properties of decomposition to predict future values. As a result, NLinear is designed to learn perturbations of future values with respect to the source. In highly non-stationary environments, the learned perturbations are minimal, causing the model to predict values that are close to the last input value.

Therefore, we shift our focus to the market data, which is a challenging TSF problem, and examine its characteristics further.

2.2 WHY DOES TRANSFORMER STRUGGLE WITH NON-STATIONARY TSF?

Transformer-based models fundamentally use Multi-Head Self-Attention, Scaled dot product attention maps, queries Q, and sets of key-value K, V pairs based on the following equation,

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V. \tag{1}$$

Multi-head attention allows the model to attend to information from different representation subspaces at different positions, enabling it to learn long-range dependencies (Vaswani et al., 2017a).

According to Liu et al. (2022), the dot-product used for attention is limited by its correlation-based similarity nature. This leads to over-stationarization, which makes the Transformer vulnerable to non-stationary time series data. According to Zhou et al. (2020) and Tsai et al. (2019), it is evident that the attention of the i-th query can be defined as a kernel smoother in a probability form as follows:

$$\mathcal{A}(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = \mathbb{E}_{p(k_j|q_i)}[v_j], \tag{2}$$

where k is the asymmetric exponential kernel $\mathbf{SM}(q, k) = \exp(q_i k_j^\top / \sqrt{p})$ (Choromanski et al., 2020), which exponentially increases based on the magnitude of the dot product similarity between q and k, leading to the over-stationarization. This is a limitation due to the nature of correlation-based similarity.

Further with deep analysis, as noted in Song et al. (2021), Multi-Head Attention follows spectral density and can be replaced by RBF kernel (Chen et al., 2021).

Lemma 2.1 ***SM***$(\cdot, \cdot)$ *is a Power Spectral Density (PSD) kernel function. Equivalently, for all integers* $n \geq 1$ *and elements* $\{\mathbf{q}_i\}_{i=1}^n \in \mathbb{R}^p$ *the n-by-n matrix* $\mathbf{C} =$ ***SM***(Q, Q) *is PSD (Chen et al., 2021).*

proof. ***SM***$(\mathbf{q_i}, \mathbf{q_j}) = exp(\frac{q_i^\top q_j}{\sqrt{p}}) = exp(\frac{||q_i||^2}{2\sqrt{p}}) exp(-\frac{||q_i - q_j||^2}{2\sqrt{p}}) exp(\frac{||q_j||^2}{2\sqrt{p}})$

Due to the dual property given by Bochner theorem (Reed & Simon, 1975) and with Monte Carlo (MC) integration, the kernel can be defined as follows (Song et al., 2021),

$$\begin{aligned} k(x, x\prime) &= \int_{\mathcal{R}^d} \exp(iw^\top (x - x\prime)) p(w) dw \\ &\approx \frac{1}{R} \sum_{r=1}^{R} \exp(iw_r^\top (x - x\prime)). \end{aligned} \tag{3}$$

As proposed in Song et al. (2021), there are various alternatives to such Gaussian kernel (Equation 3). This study demonstrates that by approximating existing attention mechanisms with kernel functions that have different properties for specific datasets, better performance can be achieved. For example, Autoformer (Wu et al., 2021) and FEDformer (Zhou et al., 2022) achieved significant performance gains by extending the attention mechanism with Fourier attention and Discrete Wavelet transform. However, these approaches only address stationarity, and due to the nature of attention as it is bounded to the Gaussian kernel, Transformer-based models are still not appropriate for non-stationary TSF. Therefore, we aim to solve the non-stationary TSF problem using a completely new approach, Diffusion, without using attention or decomposition in the encoder.

3 PRELIMINARY

Forward Process

The diffusion process consists of two main steps, which are forward and backward processes. The multivariate time series $X = \{x_{1:C,1:L}\}$, has C channels, and L length of the sequence. The forward process q is presented with Gaussian Normal noise, which is sequentially added to the original input. This can be represented as an autoregressive process of T steps:

$$\begin{aligned} q(x_{1:T}|x_0) &\coloneqq \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ q(x_t, x_{t-1}) &\coloneqq \mathcal{N}\left(\sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}\right). \end{aligned} \tag{4}$$

To control the forward process, diffusion models use a positive constant β_t that represents a noise level. Since there is a need to have a representation of the noisy value at every step of the forward process, the reparameterization technique is introduced. During this process, x_t is parameterized with the help of the x_0, the cumulative product of $\hat{\alpha}_t \coloneqq 1 - \beta_t$, and $\alpha_t \coloneqq \prod_{i=1}^{t} \hat{\alpha}_i$. This allows representation of $x_t = \sqrt{\alpha_t}x_0 + (1-\alpha_t)\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Reverse Process

The reverse process takes the final result of the forward process x_T and processes it to achieve the initial input.

$$\begin{aligned} p_\theta(x_{0:T}) &\coloneqq p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad x_T \sim \mathcal{N}(0, \mathbf{I}), \\ p_\theta(x_{t-1}|x_t) &\coloneqq \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)\mathbf{I}). \end{aligned} \tag{5}$$

Every step of the reverse process is represented with the Gaussian noise with the parameterized function of mean μ_θ and standard deviation σ_θ. To reach a more tractable representation, DDPM (Ho et al., 2020a) proposes a specific parameterization of $p_\theta(x_{t-1}|x_t)$:

$$\begin{aligned} \mu_\theta^{\text{DDPM}}(x_t, t) &= \frac{1}{\alpha_t}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\epsilon_\theta(x_t, t)}\right), \\ \sigma_\theta^{\text{DDPM}}(x_t, t) &= \tilde{\beta}_t^{1/2}, \\ \tilde{\beta} t &= \begin{cases} \frac{1-\alpha t-1}{1-\alpha_t}\beta_t & t > 1 \\ \beta_1 & t = 1. \end{cases} \end{aligned} \tag{6}$$

The following parameterization allows defining denoising function ϵ_θ, outputs of which are then compared to the originally added noise in the optimization objective:

$$\min_\theta \mathcal{L}(\theta) \coloneqq \min_\theta \mathbb{E}_{x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0, \mathbf{I}), t} \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2. \tag{7}$$

4 TPATCH-DIFFUSION

According to all aforementioned factors, we propose a novel model *TPatch-Diffusion*, which is aimed at solving all the aforementioned problems of both deterministic and probabilistic models.

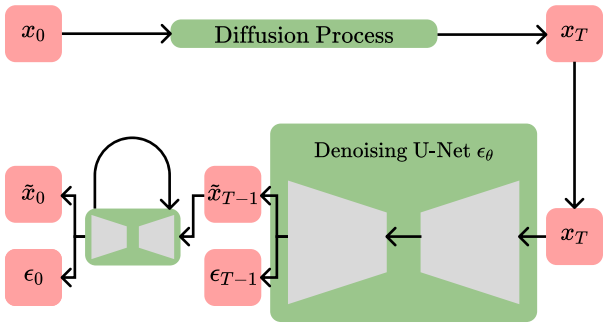


Figure 2: DDPM module architecture

It consists of two essential modules *DDPM* encoder, and *NTPatch* decoder. The former is targeted to capture the noise added to the input data, while the focus of the latter is to extract the from the model, and try to learn meaningful representations from it.

4.1 DDPM

To enable learning from noise technique, we modify a traditional DDPM model. Given the data x_0 is 1-dimensional time-series, we set all the U-Net's convolutional layers to be 1-dimensional, while excluding pooling layers, to fit the data. Additionally, we empirically found that using $tanh$ activation function instead of the sequence of batch normalization and ReLU results in better model performance, which is why we substitute them in all the inner blocks of the model.

Therefore, this module adds predefined noise value ϵ to the original data, and tries to reconstruct it with 1-dimensional U-Net. Predicted noise values are then output from the model.

Having produced added noise values, similar to the DDPM's optimization objective. We leverage L1Loss to match predicted noise values with the originally added. Therefore, optimization objective is as follows:

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\epsilon - \epsilon_\theta(x_t, t)\right]$$

where ϵ is added noise, and the $\epsilon_\theta(x_t, t)$ – denoising function.

4.2 NTPatch

Another essential module, is the *NTPatch*. This module represents the deterministic part of the model. There are two basic flows of the model, which are combined with residual connections. The first one focuses on selecting necesary information from the noisy inputs, while the second one finalizes the overall result.

The former controls the noise flow. It accepts outputs of the U-Net – ϵ_0, and segments them into patches $\epsilon_{0,[1:k]}, \ldots, \epsilon_{0,[N-k:N]}$, where k – patch size, and N – sequence length. Each patch is then processed with a separate block of layers. Every block consists of Positional Encoding, Linear layer and Softmax activation. Having all blocks produced outputs, they are combined with GRU to achieve the cumulative noise information.

Given processed noise information, it is then negated from the noisy inputs $\tilde{x}_0$, which aims to reproduce and modify the representation of original values. This representation is then projected to the output space, where cumulative noise is added to the projection to introduce uncertainty to the model outputs. This indirectly proves that Diffusion model learns noise in time series latent space. Optimization process requires both independent, and combined execution. As the TPatch-Diffusion consists of DDPM encoder and NTPatch decoder, they need additional optimization. Since DDPM model is a diffusion process, model is aimed to match predicted noise value with the original. To do so, L1Loss is utilized, which is why, it follows DDPM optimization objective mentioned in the Section: 4.1.

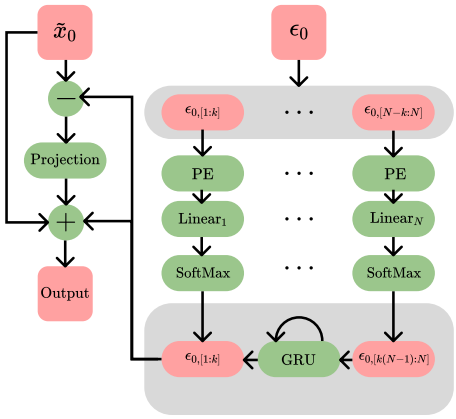


Figure 3: NTPatch module. The model accepts noisy input, and the noise. Processes the latter and sums them together to produce the final output.

NTPatch decoder loss is Mean Squared Error (MSE) Loss between produced results, and the target forecasting values. Therefore, decoder loss can be depicted in the following manner:

$$\mathcal{L}_{\text{NTPatch}}(\theta) = \mathbb{E}_{x_t,\epsilon_t}\left[\|y - y_\theta(x_t, \epsilon_t)\|^2\right]$$

Finally, global model loss is the combination of KL-divergence between the predicted and the original noise, and MSE between predictions and targets:

$$\mathcal{L} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] + \mathbb{E}_{x_t,\epsilon_t}\left[\|y - y_\theta(x_t, \epsilon_t)\|^2\right]$$

Having separate optimization steps, allows to train both independently and in combination.

5 Experiments

5.1 Experimental Setting

Baselines. To analyze the performance of the model, six different baselines were chosen. They are LTSF-Linear (Zeng et al., 2022), PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), Autoformer (Wu et al., 2021), Repeat (Zeng et al., 2022).

- `Autoformer`: As a Transformer-based method, Autoformer learns the temporal pattern of time series by decomposition and Auto-Correlation mechanism through Fast Fourier Transform (Wu et al., 2021).
- `FEDformer`: As a Transformer-based method, FEDformer introduced a Mixture of Experts (MOE) for seasonal-trend decomposition and frequency-enhanced block/attention with Fourier or Wavelet Transform (Zhou et al., 2022).
- `DLinear`: Only using Linear layers, DLinear decomposes the original input into a trend and remainder components. Then, two linear layers are applied to components respectively, and the output features are summed up to obtain the final prediction (Zeng et al., 2022).
- `NLinear`: To overcome the train-test distribution shift in the dataset, NLinear uses a simple normalization that subtracts the last value from the input and adds it back before making the final prediction (Zeng et al., 2022).
- `PatchTST`: As a transformer-based model, PatchTST has two components: segmentation of time series into subseries-level patches, and channel-independence structure. PatchTST can capture local semantic information and benefit from longer look-back windows (Nie et al., 2023).

Datasets. To evaluate the model, five non- and trend-stationary datasets were selected.

Dataset		TPatch-Diffusion (ours)		DLinear		NLinear		PatchTST		FEDformer-f		FEDformer-w		Autoformer		Repeat	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange Rate	96	0.086	0.213	0.085	0.206	0.089	0.208	0.082	0.199	0.148	0.278	0.139	0.276	0.197	0.323	**0.081**	**0.196**
	192	0.170	0.302	**0.160**	0.294	0.180	0.300	0.174	0.296	0.271	0.380	0.256	0.369	0.300	0.369	0.167	**0.289**
	336	**0.277**	**0.397**	0.328	0.424	0.331	0.415	0.351	0.425	0.460	0.500	0.426	0.464	0.509	0.524	0.305	0.396
	720	**0.468**	**0.525**	0.558	0.573	1.033	0.780	0.847	0.689	1.195	0.841	1.090	0.800	1.447	0.941	0.823	0.681
SPY	30	0.278	0.247	0.421	0.469	0.284	0.234	0.259	0.210	**0.233**	**0.192**	0.233	0.192	0.250	0.209	0.432	0.243
	60	0.331	0.263	0.663	0.596	0.310	0.265	0.289	0.241	**0.264**	**0.222**	0.264	0.222	0.286	0.240	0.488	0.278
QQQ	30	0.272	0.255	0.378	0.438	0.275	0.233	0.250	0.209	**0.228**	**0.192**	0.228	0.192	0.244	0.206	0.415	0.243
	60	0.598	0.564	**0.593**	**0.556**	0.284	0.234	0.282	0.237	0.280	0.235	0.258	0.222	0.280	0.235	0.473	0.277
MSFT	30	**0.410**	**0.449**	0.505	0.518	0.610	0.464	0.662	0.519	1.041	0.781	1.041	0.781	0.985	0.757	0.915	0.559
	60	**0.660**	0.586	0.794	0.657	0.689	0.547	0.699	**0.545**	1.423	1.027	1.423	1.027	1.337	0.958	0.974	0.572
NFLX	30	**0.785**	**0.585**	2.006	1.036	0.427	0.452	0.345	0.410	0.331	0.392	0.331	0.392	0.414	0.454	0.436	0.376
	60	**1.343**	**0.842**	3.462	1.372	1.482	0.880	1.363	0.878	0.363	0.417	0.363	0.417	0.563	0.530	0.481	0.449

Table 2: Quantitative Experiment results.

- `Exchange Rate`: The collection of the daily exchange rates of eight foreign countries including Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore ranging from 1990 to 2016 (Lai et al., 2018).
- `SPY and QQQ`: ETF stock datasets, which follows S&P500 and NASDAQ market correspondingly.
- `MSFT and NFLX`: Individual stock datasets, of Microsoft and Netflix respectively.

5.2 Performance Analysis

Quantitative Analysis
First, we compared the performance of TPatch with other state-of-the-art TSF methods through quantitative evaluation. On the Exchange rate dataset, TPatch demonstrated excellent performance in long-term prediction, while DLinear outperformed in the short term. When comparing with more non-stationary real-market data such as SPY and MSFT, TPatch-Diffusion showed the best performance except for QQQ dataset. However, the trend-stationarity property of the dataset matches the performance of existing models with TPatch-Diffusion. Finally, it is important to mention that unlike existing models, TPatch-Diffusion successfully solves long-term forecasting problems, since its performance does not plummet with the increase of prediction horizon.

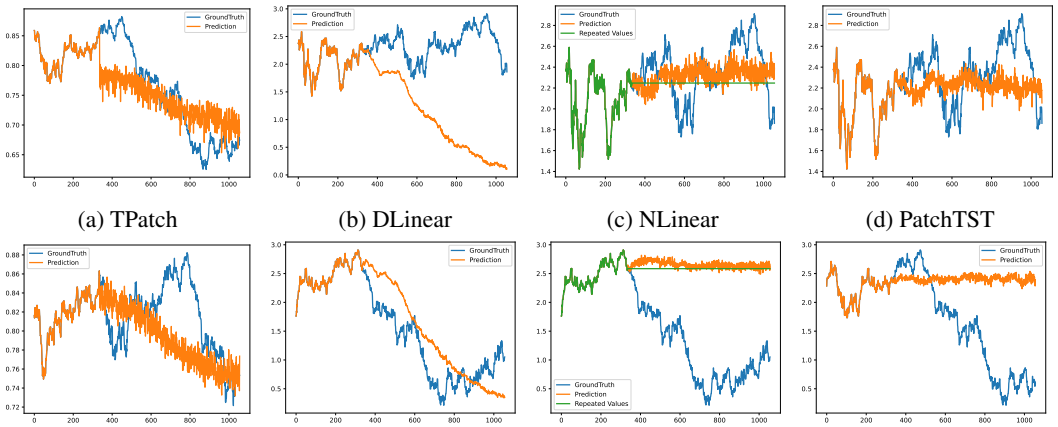
Although, FEDformer and Autoformer models show better quantitative performance, qualitatively they learn single pattern, which is considered to be optimal, and predict it for all cases. The reason for this may be frequency decomposition technique used by the models.

Qualitative Analysis
DLinear and NLinear models both belong to the LTSF-Linear family. DLinear decomposes the data into trend and seasonality fragments and successfully captures the trend, as evidenced by the graph (f) in Figure 4. However, DLinear's heavy reliance on the trend, leads to significantly different predictions when the trend of the actual data changes, as shown in (b).

On the other hand, the NLinear model learns based on the last values and introduces noise for perturbation. Consequently, it only captures small fluctuations and struggles to predict future values in non-stationary setting, as observed in the graphs (c, g) of Figure 4. The same limitation applies to PatchTST, which is a transformer-encoder-based model that divides the time-series into patches.

6 Conclusion

In this study, we address the problem of non-stationary time series forecasting using a diffusion model, which is a representative generative model. Traditional Transformer-based TSF models are designed with a decomposition mechanism to perform well on stationary data. Motivated by this, we analyze the stationarity of existing TSF benchmark datasets and find that the majority of these datasets consist of stationary data. Therefore, we specifically focus on using stock data, which is a prominent example of non-stationary time series data. TPatch-Diffusion learns the distribution of time series data through probabilistic learning in a generative model framework and enables non-stationary time series prediction through the denoising process of DDPM. We verify that TPatch-Diffusion demonstrates superior performance on market datasets, highlighting the difficulty of predictions for non-stationary data when existing TSF models excessively rely on decomposition methods.



(a) TPatch (b) DLinear (c) NLinear (d) PatchTST

(e) TPatch (f) DLinear (g) NLinear (h) PatchTST

Figure 4: Qualitative Experiments results on Exchange Rate dataset.

6.1 Future Work

We plan to directly analyze the impact of generative models on non-stationary time series forecasting. Firstly, through a generative model ablation study, we will analyze the differences when using VAE, GAN, and Diffusion models. Additionally, we aim to find the justification for using the diffusion model through this process. Furthermore, we will analyze stock data from the perspective of Brownian motion to provide evidence for why this data is non-stationary and why generative models are necessary.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL https://arxiv.org/abs/2004.05150.

Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nystr\" om method. *Advances in Neural Information Processing Systems*, 34:2122–2135, 2021.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

David A Dickey and Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020a. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020b.

Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting, 2022.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Michael Reed and Barry Simon. *II: Fourier Analysis, Self-Adjointness*, volume 2. Elsevier, 1975.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Kyungwoo Song, Yohan Jung, Dongjun Kim, and Il-Chul Moon. Implicit kernel attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9713–9721, 2021.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: A unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017a. URL http://arxiv.org/abs/1706.03762.

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 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. *CoRR*, abs/2012.07436, 2020. URL https://arxiv.org/abs/2012.07436.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *CoRR*, abs/2201.12740, 2022. URL https://arxiv.org/abs/2201.12740.