Trend learning based loss function for time-series forecasting

Haibin Liao[a,b], **Yiyang Hu**[a], **Li Yuan**[a]

[a]School of Electronic and Electrical Engineering, Wuhan Textile University, Wuhan,430200 China
[b]Hubei Province Engineering Research Center for Intelligent Micro-nano Medical Equipment and Key Technologies
liao_haibing@163.com, 3184827268@qq.com, 2022048@wtu.edu.cn

Abstract

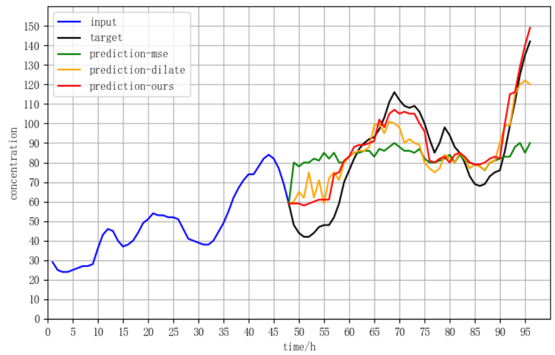
The loss function is a vital component of time-series forecasting models based on deep learning. The widely used L_p norm distances-based loss functions, which is vulnerable to not consider dynamic temporal patterns, resulting in its inability to capture the trend or shape of the sequence well. On the other hand, in practice, the sequence data will have various distortions due to the influence of environmental factors, and how to overcome the data distortion variations is an important challenge for loss function design in time series forecasting. Aiming at the above problems, we design a loss function framework based on time-series trend transformation, which consists of trend direction guidance and point-wise representation terms. Based on the loss function framework, we propose a novel loss function, called Tre-Loss (Trend learning based loss function), that not only considers the distortions in all aspects but also allows models to capture the trend or shape of time-series. We evaluate the effectiveness of Tre-Loss by conducting extensive experiments from naive models to state-of-the-art models. The experiment results indicate that the models trained with Tre-Loss outperforms those trained with other training metrics (e.g., MSE, DILATE and TILDE-Q). Our code and models will be available at GitHub https://github.com/liaohaibing/Trend-learning-loss-function.

1 Introduction

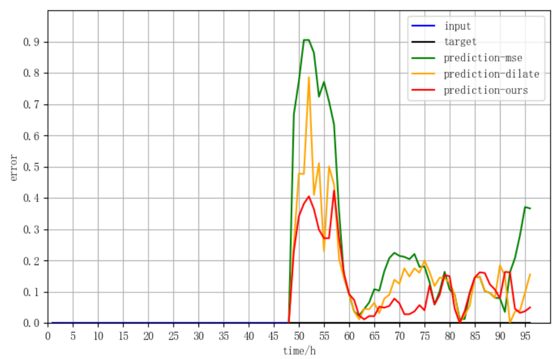
Time-series data are becoming ubiquitous in numerous real-world applications, e.g., wearable devices, IoT devices, smart city, financial markets, environmental sciences, renewable energy sciences, etc. Given the availability of a large number of data, their complex underlying structures/distributions, together with the high-performance computing platforms, there is a great demand for developing new theories and algorithms to tackle fundamental challenges (e.g., representation, forecasting, reconstruction, mining analysis, etc.) in various types of applications.

In the construction of smart cities, time-series forecasting is particularly important. In the past, linear fitting/regression models were mainly used for time-series prediction [Box et al., 2015; Durbin and Koopman, 2012]. Linear model methods have simplicity and interpretability, but they rely too much on prior knowledge (such as periodicity and seasonality), resulting in their inability to predict non-equilibrium series, mutability and long time-series. Deep learning models are an appealing solution for this problem [Yu et al., 2017; Zhou et al., 2021], due to their automatic feature extraction and complex nonlinear time dependencies modeling. In deep learning methods, the design of the loss function (evaluation criteria) is very important when dealing with complicated time-series prediction problems, because it can guide the model to learn the best state correctly and quickly.

At present, most of the time-series forecasting models based on deep learning are trained using Mean Squared Error (MSE) or its variants (MAE, quantile loss, etc.) as a proxy loss function. However, the MSE has fatal defect for evaluation sequence forecasts, especially in non-stationary environment with drastic changes [Vallance et al., 2017; Verbois et al., 2020]. MSE pays too much attention to prediction error and ignores time-series' temporal dynamics, such as rise, drop, trough, peak, and plateau. The time-series forecasting model should not only pursue the accuracy of numerical prediction in each time-step, but also make the predicted series have similar trends to the real series. However, most of existing models do not consider learning trend, so the forecasting results are often inaccurate and uninformative, because deep learning model tends to learn in an easy way. Figure 1 shows three real forecasting results with same model, different loss functions. When we use MSE as a loss function, the model only dedicated to reduce gap between prediction and ground truth for each time-step. That is, the model only producing relatively simple predictions regardless trend information. A negative result of this is that the model will generate relatively large prediction errors. (Figure 1 (b)). Based on this, if we consider both value gap and sequential trend of forecast and ground truth, trends will guide value prediction, resulting in the accuracy and shape could both improved, which as shown in Figure 1 (a).



(a) three real forecasting results with same model, different loss functions



(b) three forecasting error with same model, different loss functions

Figure 1: Ground-truth and predicted results with three metrics (MSE, DILATE, and ours)

Due to the unique temporal pattern continuity of time-series, MSE is only the embodiment of different sequences in specific values, and contains too little information, which cannot well reflect the trend changes and distortions of sequences [Lee et al., 2023]. To better learning temporal dynamics in time-series, differentiable, approximated dynamic time warping (DTW) [Le Guen et al. 2023; Frías-Paredes et al., 2017], have used as an alternative metric of MSE. At present, most advanced loss function designs in time-series prediction models are considered from two aspects: shape similarity calculation and temporal error estimation [Le Guen and Thome, 2019]. For shape similarity calculation, the ramp score [Florita et al., 2013; Vallance et al., 2017] is proposed based on a piecewise linear approximation of the derivatives of time-series; In order to make the predicted sequence as similar as possible to the shape of the ground-truth sequence, scholars designed some time-series evaluation criterions based on DTW [Cuturi and Blondel, 2017, Abid and Zou, 2018; Frías-Paredes et al., 2017]. For temporal error estimation, the Temporal Distortion Index (TDI) based on the DTW algorithm – which quantifies the temporal distortion between two time series [Frías-Paredes et al., 2017]. However, using DTW as a loss function results in ignoring temporal localization of changes. To solve this problem, Le Guen & Thome [2019] suggests DILATE, a training metric to timely catch sudden changes of non-stationary signals with smooth approximation of DTW and penalized temporal distortion index (TDI). To guarantee to work in a timely manner, they introduce a loss function that gives a harsh penalty when predictions show high temporal distortion. However, DILATE often loses its advantage with complex data because of DTW often shows misalignment. DTW or its variants construct the correspondence between the predicted sequence and the ground-truth sequence according to the principle of nearest distance, and evaluate the similarity of the two sequences, while TDI relies on the DTW path. In order to find the minimum error when calculating error, DTW allows to find the best matching point near the calculation point, which relaxes the requirement of model learning and destroys the one-to-one correspondence between sequences. DTW is used well in the field of speech recognition (allowing variations in delay, interruption, and speed). TDI has good applications in the field of time-sensitive renewable energy forecasting such as wind and solar. Recently, in order to better learn the shape and time patterns of sequences, Lee et al. [2023] proposed a loss function TILDE-Q, which allows amplitude and phase offset between sequences in the loss function design.

In this work, we aim to develop a novel loss function that to better learn the dynamic temporal patterns and cope with the distortion of sequences by introducing trends similarities and dissimilarities for training deep forecasting models in time-series data.

2 Methods

2.1 Notations and Definitions

Definition 1 *(Time-series forecasting)**.** Given τ_{in} -length historical time-series $X=\left[X_{t-\tau_{in}+1}, X_{t-\tau_{in}+2}, \cdots, X_{t}\right]$, $X_i \in \mathbb{R}^F$, at time i ; time-series forecasting task is use the learned mapping function $f: \mathbb{R}^{\tau_{in} \times F} \rightarrow \mathbb{R}^{\tau_{out} \times C}$ to predict τ_{out}-length future time-series $Y=\left[Y_{t+1}, Y_{t+2}, \cdots, Y_{t+\tau_{out}}\right], Y_i \in \mathbb{R}^{\tau_{out} \times C}$.*

Assume the label data（i.e., ground-truth）as Y and prediction data as $\hat{Y}$, the time-series forecasting goals aims to require not only precise, but also informative forecasting [Zhou et al., 2021, Lee et al., 2023, Le Guen et al., 2023] as: 1) Mapping function f should be learnt to point-wisely reduce distance between Y and $\hat{Y}$; 2) The output $\hat{Y}$ should have similar trending or shape dynamics with Y .Trending dynamics are informative patterns in time-series, such as fall, rise, peaks and valleys etc. In this work, we aim to design a trend-aware loss function that satisfies both goals. To this end, we first discuss distortions that two time-series with similar trends can have.

Definition 2 *(Time-series distortion)**.** Given two time-series Y and $\hat{Y}$ in a similar trend or shape, distortion is a difference between Y and $\hat{Y}$. That is, Y and $\hat{Y}$ belong to the same time-series, but they are different due to distortions.*

Distortions are mainly from value distortion (i.e., scaling) and temporal distortion (i.e., warping) with respect to its relevance of dimension, time and amplitude, which including time_offset, value_offset, scaled, noise and time_warped.

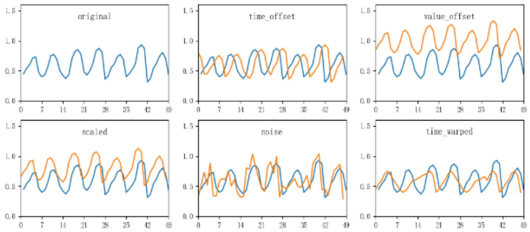


Figure 2. Example of the five distortions on the amplitude axis and temporal axis.

Figure 2 presents example distortions, categorized by amplitude and time dimensions:

- **time_offset** can be represented with two time-series functions with the degree of offset (k): $\hat{Y}(t)=Y(t+k)$, where $k\in\mathbb{R}$ is constant.
- **value_offset** can be described with two time-series and the degree of offset (k): $\hat{Y}(t)=Y(t)+k$,where $k\in\mathbb{R}$ is constant.
- **scaled** is a transformation that changes the amplitude or temporal by multiplication of $k\in\mathbb{R}$. This distortion can be described with two functions and a multiplication factor (k): $\hat{Y}(t)=k\cdot Y(t)\ or\ \hat{Y}(t)=Y(k\cdot t)$.
- **noise** can be interpreted as any distortion occurred by non-zero add on the amplitude dimension. This distortion can be described as: $\hat{Y}(t)=Y(t)+\delta$, where δ is a random value.
- **time_warped** means any transformation that dynamically lengthens or shortens signals on the temporal dimension including local time scaling and occlusion. It can be represented as follows: $\hat{Y}(t)=Y(h(t))$,where $h(\cdot)$ represents the time transformation function.

2.2 Loss function design of time-series forecasting model

A vital component of a machine learning or deep learning model used for time-series forecasting is the loss function. The design of loss function is mostly derived from the time-series similarity measurement criteria. A model's performance is measured against the loss, and the parameters it chooses to learn will be based on the minimization of function. The measurement metrics widely used in the field of time-series forecasting are all based on L_p distance, the corresponding loss function is [Jadon et al., 2022]:

$$L_p(y_t,\hat{y}_t)=\frac{1}{T}\sum_{t=1}^{T}\left|y_t-\hat{y}_t\right|_p \tag{1}$$

Distance-based L_p is a point-wise mapping method, which has good robustness to time-series noise. However, value_offset, scaled distortion has a large impact on it; At the same time, it cannot handle temporal distortions appropriately and vulnerable to scaling of the data, so, it is also sensitive to time_offset and time_warped distortions.

2.3 Training time-series forecasting model with Tre_Loss

Unlike all previous loss function designs, considering the trend of time-series, we propose a loss function framework based on time-series trend transformation:

$$L_F=\lambda L_{guide}+(w_{tre}\cdot L_{value}) \tag{2}$$

Where, λ is the weight parameters; L_{guide} indicates the trend direction; L_{value} is the point-wise representation item; w_{tre} is the trend weight, which used for supervised learning of L_{value}. Therefore, we design a loss function framework as a combination of directional guidance terms and specific value performance items. The directional guidance item guides the model to learn the correct transformation trend to avoid deviating from the direction; The specific value performance term is the final desired result, and in order to ensure the accuracy of the prediction results, we use trend weights for supervised learning. Through this loss function framework, all the five time-series distortions mentioned above can be unified into this framework. Whether it's time_offset, value_offset, scaled, noise and time_warped, they're all trend-keeping problems that can be described in terms of trend changes.

Based on this loss function framework, we can design a new loss function (Tre-Loss):

$$L_{tre}=\lambda\left|fft(TR[\hat{y}])-fft(TR[y])\right|+\left(\frac{1}{T}\sum_{t=1}^{T}\left(1.01+\frac{|\hat{y}_t-y_t|}{|\hat{y}_t+y_t|}\right)^{(1-\delta(t))}|\hat{y}_t-y_t|\right) \tag{3}$$

Where, T is the forecast duration of the time-series, y is the ground-truth value, $\hat{y}$ is the predicted value; $fft(\cdot)$ represents the Fourier transform, $TR[\cdot]$ represents the trend vector of the sequence, $\delta(t)$ represents the trend consistency judgment function. $TR[\hat{y}]$ and $TR[y]$ respectively represent the trend change vectors of the predicted sequence and the real sequence within a given time T, whose values are composed of second-order discrete partial derivatives at different specific time points. The second-order discrete partial derivatives processed by time t are calculated as follows:

$$TR[y_t]=(y_t-y_{t-1})+(y_{t+1}-y_t)=(y_{t+1}-y_{t-1}) \tag{4}$$

The trend consistency judgment function $\delta(t)$ is:

$$\delta(t)=\begin{cases}1 & if\,(y_t-y_{t-1})*(\hat{y}_t-\hat{y}_{t-1})>0\\ 0 & if\,(y_t-y_{t-1})*(\hat{y}_t-\hat{y}_{t-1})=0\\ -1 & if\,(y_t-y_{t-1})*(\hat{y}_t-\hat{y}_{t-1})<0\end{cases} \tag{5}$$

t represents the time step. In Eq. (3), the first item is the trend direction guide item, and the second item is the specific value representation item, the trend weight w_{tre} can be regarded as:

$$w_{\text{tre}} = (1.01 + \frac{|\hat{y}_t - y_t|}{|\hat{y}_t + y_t|})^{(1-\delta(t))} \quad (6)$$

The first term of Tre_Loss is the trend transformation loss, which forces the predicted sequence to have a consistent trend with the ground-truth sequence at each time point t as much as possible; The second term is the weighted MAE loss function. If the predicted trend at a certain moment is the same as the ground-truth trend, it degenerates into MAE; If the predicted trend is different from the ground-truth trend, increase its weight to attract more attention. For example,

True: the true sequence is 10-11-16

Forecasting 1: prediction sequence 1 is: 10-8-#

Forecasting 2: prediction sequence 2 is: 10-15-#

Given the choice between two prediction sequences, it is clear that we prefer the **Forecasting 2** because the trend in the second is consistent with the trend in the true sequence. Although 8 in **Forecasting 1** is only 3 different from the true 11, and 15 in **Forecasting 2** is 4 different from the true 11, the trend of **Forecasting 2** is consistent with the true trend, while the trend of **Forecasting 1** is opposite, which brings difficulties to its subsequent prediction. If **Forecasting 1** wants to predict 16 in the third step, it needs to double the result predicted in the second step, that is, mutations are generated. However, in the prediction of time-series, mutations are either accidental events or abnormal data, which is random and difficult to predict accurately. In contrast, **Forecasting 2** predicts 16 in step 3 only with a mild change in the normal trend.

3 Experiments

In order to verify the validity of the proposed Tre_Loss, we conduct comprehensive experiments with different datasets, demonstrate its effectiveness and the distortion robustness characteristics.

3.1 Datasets and Setup

We use five datasets: AQI, Traffic, ETT, ECL, and Synthetic dataset:

AQI dataset: is air quality data set from different monitoring stations in Jinan City collected by us in projects. There are 130 air monitoring stations in Jinan. Each station outputs the concentration values of pollutants (PM2.5, PM10, SO2, CO2, CO, O3) and meteorological parameters every hour. We collected the historical monitoring data of 130 stations from January 1st, 2019, to January 1st, 2022, as the training and test set.

Traffic dataset: is hourly road occupancy rate data set collection in the California Department of Transportation from 2015 to 2016. We conducted experiments using a univariate sequence of the first sensor, with a total of 17544 data points. We split the dataset into training data, validation data, and test data by the radio of 0.6:0.2:0.2.

ETT dataset [Zhou et al., 2021]: is multiple datasets involving power transformers. All data is preprocessed and stored in.CSV format. The data covers the period from July 2016 to July 2018, we use the High UseFul Load (HUFL) in ETTh2 as the experiment object, and split the dataset into training data, validation data, and test data by the radio of 0.6:0.2:0.2.

ECL dataset [Zhou et al., 2021]: is 2-year electricity consumption (Kwh) data of 1-hour intervals collected from 321 clients. We split the dataset into training data, validation data, and test data by the radio of 0.68:0.14:0.18. Detailed settings are based on the information at Informer Github[1].

Synthetic dataset: is an artificial dataset for measuring model performance on sudden changes with an input signal composed of two peaks. The amplitude and temporal position of the two peaks are randomly selected. We generation 500 sequences for training, 500 for validation, and 500 for testing. The generation code is provided in DILATE Github[2].

We utilize previous 48-hour observations and select the results of 48 hours ahead forecasting to report. We perform experiments with two different model architectures, including Sequence-to-Sequence (Seq2Seq) GRU and Informer [Zhou et al., 2021]. Seq2Seq GRU to evaluate Tre_Loss in simple model, we utilize one-layer Seq2Seq GRU model. For the training of the GRU model, we set learning rate of $1e^{-3}$, hidden size of 128, trained by maximum 1000 epochs with Early Stopping and ADAM optimizer. We utilize the official code and hyperparameter setting to train Informer.

To evaluate the performance of our Tre_Loss, we compare it against the widely used Euclidean loss (MSE), DILATE [Le Guen et al., 2023], TILDE-Q [Lee et al., 2023]. Refer to Le Guen & Thome [2023], we use the following multi-step prediction metrics: MAE (absolute error), SMAPE (relative error), DTW (shape), TDI (temporal). The code is released on GitHub.

3.2 Tre_Loss performances on generic architectures

To demonstrate the broad applicability of Tre_Loss, we first perform sequence forecasting with a simple Seq2Seq model with 1 layer of 128 GRU. The results are shown in Table 1.

From the table, it can be seen that Tre_Loss achieved the best overall performance in both MAE and SMAPE metrics on 5 different datasets. Furthermore, it can be seen that DTW and TDI indicators are good, but their corresponding predictive performance (MAE and SMAPE) may not necessarily be good.

[1] https://github.com/zhouhaoyi/Informer2020

[2] https://github.com/vincent-leguen/DILATE

Dataset	Eval	Train Seq2Seq+ MSE	Seq2Seq+ DILATE	Seq2Seq+ TILDE-Q	Seq2Seq+ ours
AQI	MAE	27.5286	27.3092	25.9823	**23.6546**
	SMAPE	0.5013	0.5094	0.4936	**0.4594**
	DTW	189.6806	**168.1006**	168.5467	176.5589
	TDI	9.0976	7.3966	**5.4658**	9.0157
Traffic	MAE	0.0070	0.0095	0.0072	**0.0065**
	SMAPE	0.4986	0.5328	0.5147	**0.4963**
	DTW	1.4628	1.6929	**1.4600**	1.4688
	TDI	0.2343	0.2814	0.2276	**0.2215**
ECL	MAE	329.3321	297.7066	297.1863	**284.8722**
	SMAPE	0.0953	0.0866	0.0854	**0.0829**
	DTW	1760.0057	1649.0998	1612.4682	**1521.3708**
	TDI	2.0257	1.6987	1.5367	**1.3354**
ETT	MAE	3.8388	4.6727	3.9698	**3.6878**
	SMAPE	0.1428	0.1669	0.1526	**0.1384**
	DTW	25.5583	25.4878	24.4566	**24.2588**
	TDI	**2.8033**	3.3931	3.1582	2.9016
Synthetic	MAE	0.0748	0.0490	0.0517	**0.0461**
	SMAPE	0.6159	0.5003	0.5136	**0.4887**
	DTW	0.3543	**0.1782**	0.1796	0.1787
	TDI	1.6302	1.7691	1.6953	**1.2118**

Table 1: Different loss functions comparison results with Seq2Seq and different dataset

3.3 Tre_Loss Performances with state-of-the-art models

Dataset	Eval	Train Informer+ MSE	Informer+ DILATE	Informer+ TILDE-Q	Informer+ ours
AQI	MAE	33.4947	29.6882	30.2637	**28.7834**
	SMAPE	0.6878	0.6181	0.6248	**0.6085**
	DTW	235.4214	**158.5667**	159.7469	160.6937
	TDI	**7.2632**	10.6815	11.2518	9.8639
Traffic	MAE	0.0017	0.0017	0.0017	**0.0017**
	SMAPE	0.3327	0.3309	0.3308	**0.3307**
	DTW	0.0114	**0.0067**	0.0097	0.0102
	TDI	**7.8636**	10.41619	9.7821	8.6839
ECL	MAE	**216.9933**	361.0186	298.3754	287.7382
	SMAPE	**0.0626**	0.1052	0.0974	0.09462
	DTW	1117.5870	1121.4119	**1108.2736**	1120.7362
	TDI	**0.3965**	1.3377	1.2680	1.4137
ETT	MAE	4.9867	7.1404	5.0791	**4.6758**
	SMAPE	0.1780	0.2418	0.1947	**0.1581**
	DTW	33.2952	35.4807	26.1518	**26.0681**
	TDI	5.9820	6.1822	**4.7934**	4.9815
Synthetic	MAE	0.0865	0.0582	0.0672	**0.0567**
	SMAPE	0.7158	0.6028	0.6157	**0.5869**
	DTW	0.4636	**0.2688**	0.2897	0.2785
	TDI	1.9314	1.9895	1.9963	**1.5267**

Table 2: Different loss functions comparison results with Informer and different dataset

Beyond generic forecasting architectures, we show that Tre_Loss can also improve the performances of state-of-the art deep architectures. We experiment here with Informer. Results in Table 2 are consistent with those in Table 1: Our proposed Tre_Loss achieves the best comprehensive performance in both MAE and SMAPE metrics. At the same time, we found that Informer's performance in short and medium time series prediction is inferior to that of Seq2Seq. When the training sample is insufficient, the performance improvement effect of using advanced loss function is not obvious.

3.4 Ablation study

To evaluate the effect of the λ, we conduct a set of experiments using Seq2Seq with different datasets. As we can see in the Table 3, on the datasets of AQI, Traffic and Synthetic, when $\lambda = 0$, its predictive performance is significantly worse than when $\lambda > 0$, indicating that the trend direction guidance term L_{guide} is helpful for the accuracy of prediction; Adopting $\lambda = 1$, $\lambda = 1.5$ or $\lambda = 2$ a relatively safe choice, but not the optimal one; The optimal choice needs to be obtained through cross validation based on the actual dataset.

Dataset	Eval	λ 0	0.5	1	1.5	2	3
AQI	MAE	28.22	25.74	24.91	24.85	**23.65**	24.03
	SMAPE	0.57	0.50	0.48	0.47	**0.45**	0.47
	DTW	200.53	189.68	178.99	176.98	**176.55**	178.97
	TDI	9.16	9.93	9.14	10.26	**9.01**	9.53
Traffic	MAE	0.11	0.11	0.11	**0.10**	0.11	0.12
	SMAPE	0.27	0.26	0.27	**0.25**	0.25	0.28
	DTW	0.72	0.73	0.71	**0.66**	0.70	0.77
	TDI	11.06	9.49	8.97	**9.66**	9.48	8.04
ECL	MAE	292.64	293.39	**284.87**	298.47	316.49	323.92
	SMAPE	0.08	0.08	**0.08**	0.08	0.09	0.09
	DTW	1534.60	1605.91	**1521.37**	1610.81	1666.76	1869.91
	TDI	1.38	1.53	**1.33**	1.50	1.64	2.13
ETT	MAE	3.73	4.29	3.85	3.69	4.12	**3.68**
	SMAPE	0.13	0.15	0.14	0.13	0.15	**0.13**
	DTW	24.44	26.40	25.22	24.27	26.32	**24.25**
	TDI	3.00	3.41	3.04	2.99	2.78	**2.90**
Syn-thetic	MAE	0.07	**0.04**	0.05	0.07	0.06	0.04
	SMAPE	0.61	**0.48**	0.51	0.60	0.58	0.50
	DTW	0.35	**0.17**	0.17	0.34	0.27	0.17
	TDI	1.63	**1.21**	1.69	1.45	1.21	1.76

Table 3: Ablation study results with different datasets

3.5 Qualitative analysis

In order to further illustrate the characteristics of Tre_Loss, we conduct a qualitative presentation analysis. Figure 3-7 show how the model present with different training metrics under different datasets. From the figure, we have noticed that Tre_Loss enforces the model to generate more robust, trend-aware forecasting, regardless of the time_offset, value_offset, scaled and noise. For example, under Synthetic dataset (Figure 3), Tre_Loss generates more robust, trend-aware forecasting results compared to other loss function. Even when the model lacks sufficient ability to capture sequence trends (Figure 5), Tre_Loss strives to align with the real trend. When the model has the ability to learn sequence trends (Figure 4), Tre_Loss is more robust to data distortion such as noise compared to other loss functions. In summary, Tre_Loss proves that it is distortion-robust, and able to capture the sequence trends.

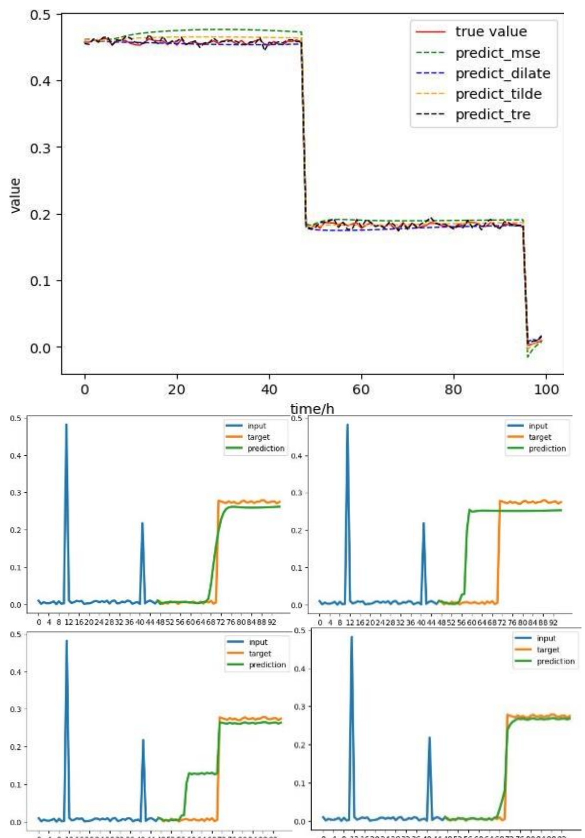


Figure 3: Qualitative results with Synthetic dataset
(The upper part of the figure compares the prediction results of four different loss functions, the four small images in the lower part show the prediction results of MSE, DILATE, TILDE-Q, and Tre_Loss, respectively)

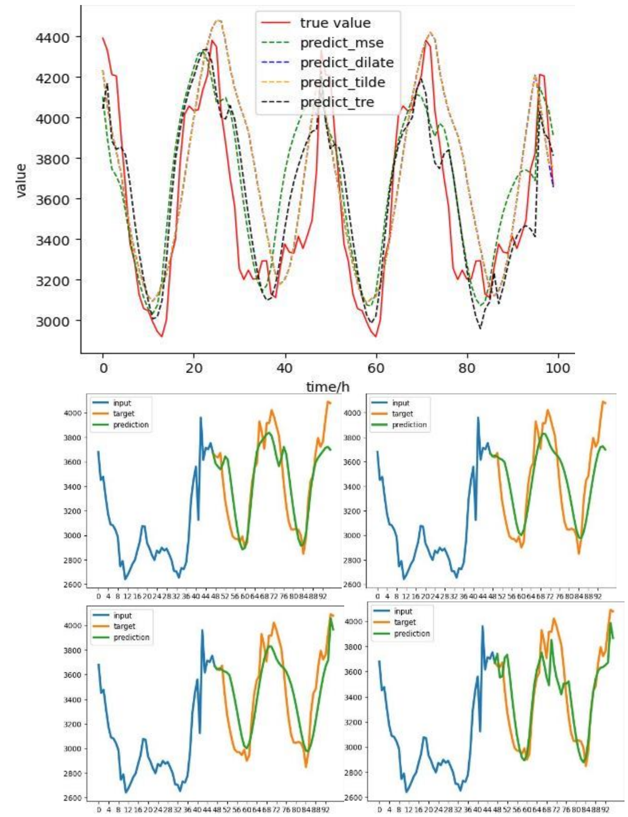
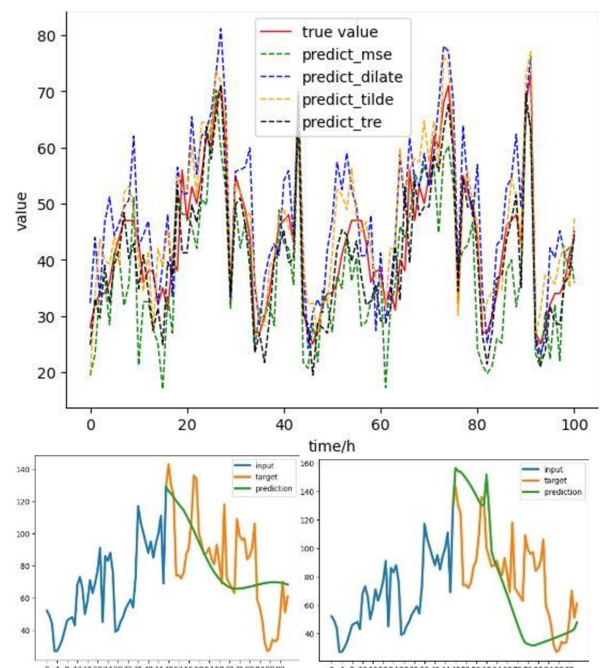


Figure 4: Qualitative results with ECL dataset



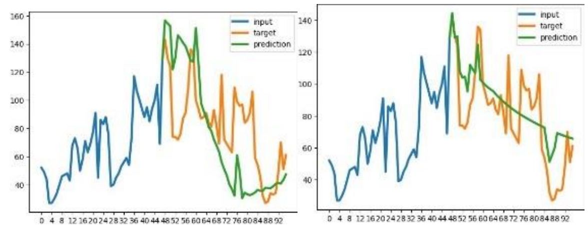
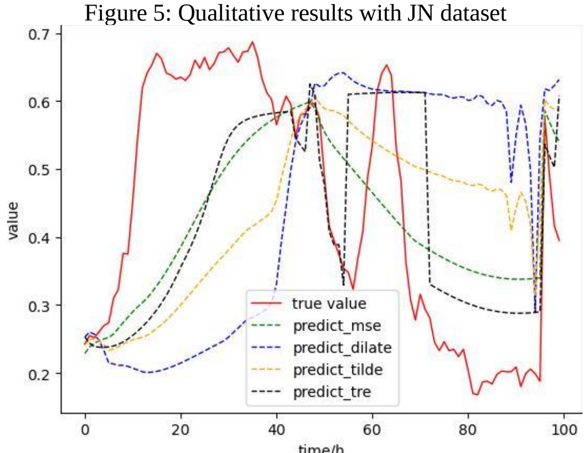


Figure 5: Qualitative results with JN dataset



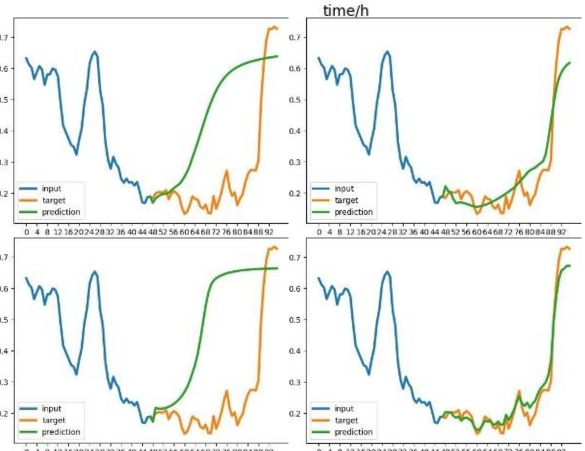
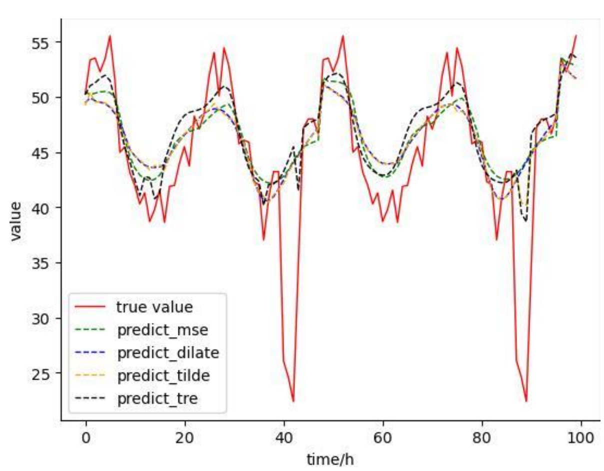


Figure 6: Qualitative results with Traffic dataset



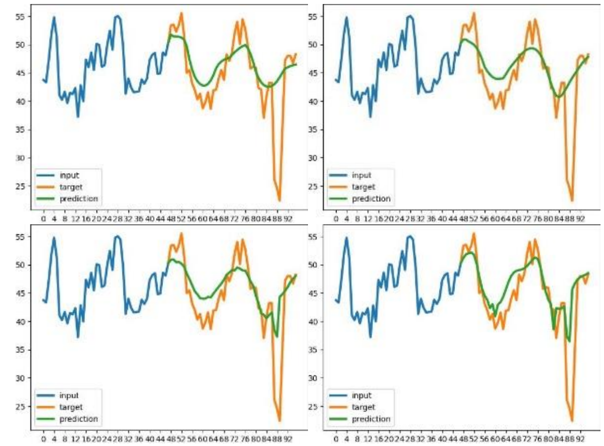


Figure 7: Qualitative results with ETT dataset

4 Conclusion and future work

In this paper, we analyze the distortion of time series and design a loss function framework based on trend learning, which can characterize the distortion variations of the series. Based on this framework, we propose Tre-Loss, a transformation invariant loss function with distance equilibrium, which allows trend-aware time-series forecasting in a timely manner. The Tre-Loss ensures a model to be invariant to the time_offset, value_offset, scaled and noise.

The study of the trend direction guideline item in the loss function framework is not comprehensive enough, and the improvement of the trend direction guideline item in Tre-Loss is not obvious enough. For future work we intend to further delve into the trend direction guidance items and explore the extension of these ideas to time series reconstruction.

References

[Box *et al.*, 2015] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control. New York, NY, USA:Wiley, 2015.

[Durbin and Koopman, 2012] J. Durbin and S. J. Koopman, Time Series Analysis by State Space Methods. London, U.K.: Oxford Univ. Press, 2012.

[Yu et al., 2017] R. Yu, S. Zheng, and Y. Liu, "Learning chaotic dynamics using tensor recurrent neural networks," in Proc. Int. Conf. Mach. Learn. Workshop Deep Structured Prediction, 2017.

[Qin *et al.*, 2017] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in Proc. Int. Joint Conf. Artif. Intell., 2017.

[Lai *et al.*, 2018] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Informat. Retrieval, 2018, pp. 95–104.

[Salinas *et al.*, 2020] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," Int. J. Forecasting, vol. 36, no. 3, pp. 1181–1191, 2020.

[Oreshkin *et al.*, 2020] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "NBEATS: Neural basis expansion analysis for interpretable time series forecasting," in Proc. Int. Conf. Learn. Representations, 2020.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12):11106–11115, 2021.

[Vallance *et al.*, 2017] L. Vallance, B. Charbonnier, N. Paul, S. Dubost, and P. Blanc, "Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric," Sol. Energy, vol. 150, pp. 408–422, 2017.

[Cuturi and Blondel, 2017] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 894–903.

[Wang and Bovik, 2009] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," IEEE Signal Process. Mag.,vol. 26, no. 1, pp. 98–117, Jan. 2009.

[Yang *et al.*, 2020] D. Yang et al., "Verification of deterministic solar forecasts," Sol. Energy, vol. 210, pp. 20–37, 2020.

[Verbois *et al.*, 2020] H. Verbois, P. Blanc, R. Huva, Y.-M. Saint-Drenan, A. Rusydi, and A. Thiery, "Beyond quadratic error: Case-study of a multiple criteria approach to the performance assessment of numerical forecasts of solar irradiance in the tropics," Renewable Sustain. Energy Rev., vol. 117, 2020, Art. no. 109471.

[Lee *et al.*, 2023] Lee, Hyunwook, Chunggi Lee2, Hongkyu Lim1, Sungahn Ko. TILDE-Q: A TRANSFORMATION INVARIANT LOSS FUNCTION FOR TIME-SERIES FORECASTING[c]. ICLR 2023, 10.48550/arXiv.2210.15050.

[Le Guen *et al.*, 2023] Le Guen, Vincent, Thome, Nicolas. Deep Time Series Forecasting With Shape and Temporal Criteria[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 45, Issue 1, Pages 342-355, January 1, 2023.

[Abid and Zou, 2018] Abubakar Abid and James Y Zou. Learning a warping distance from unlabeled time series using sequence autoencoders. In Advances in Neural Information Processing Systems, volume 31, pp. 10568–10578, 2018.

[Cuturi and Blondel, 2017] Marco Cuturi and Mathieu Blondel. Soft-dtw: A differentiable loss function for time-series. In Proceedings of the 34th International Conference on Machine Learning, ICML'17, pp. 894–903, 2017

[Frías-Paredes *et al.*, 2017] Frías-Paredes, Laura,Mallor, Fermín,Gastón-Romeo, Martín,et al.Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors[J].Energy Conversion and Management, 2017, 142:533-546.

[Le Guen and Thome, 2019] Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[Florita *et al.*, 2013] Anthony Florita, Bri-Mathias Hodge, and Kirsten Orwig. Identifying wind and solar ramping events. In 2013 IEEE Green Technologies Conference (GreenTech), pages 147–152. IEEE, 2013.

[Vallance *et al.*, 2017] Loïc Vallance, Bruno Charbonnier, Nicolas Paul, Stéphanie Dubost, and Philippe Blanc. Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric. Solar Energy, 150:408–422, 2017.

[Frías-Paredes *et al.*, 2017] Laura Frías-Paredes, Fermín Mallor, Martín Gastón-Romeo, and Teresa León. Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors. Energy Conversion and Management, 142:533–546, 2017.

[Vallance *et al.*, 2017] Loïc Vallance, Bruno Charbonnier, Nicolas Paul, Stéphanie Dubost, and Philippe Blanc. Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric. Solar Energy, 150:408–422, 2017.

[Frías-Paredes *et al.*, 2016] Frías-Paredes L, Mallor F, León T, Gastón-Romeo M. Introducing the Temporal Distortion Index to perform a bidimensional analysis of renewable energy forecast. Energy 2016;94:180–94.

[Frías-Paredes *et al.*, 2017] Laura Frías-Paredes, Fermín Mallor, Martín Gastón-Romeo, Teresa León,Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors, Energy Conversion and Management, Volume 142, 2017,Pages 533-546.

[Jadon *et al.*, 2022] Aryan Jadon, Avinash Patil, Shruti Jadon, A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting, 2022,arXiv:2211.02989, https://doi.org/10.48550/arXiv.2211.02989