Data quality-based gradient optimization for recurrent neural networks

Feihu Huang
School of Computer Science
Civil Aviation Flight University of China
Guang Han, China

Peiyu Yi
College of Computer Science
Sichuan University
Cheng Du, China

Shan Li*
National Science Library(Chengdu)
Chinese Academy of Sciences
Cheng Du, China

Haiwen Xu
School of Computer Science
Civil Aviation Flight University of China
Guang Han, China

ABSTRACT

Time series forecasting holds significant value in various application scenarios. However, existing forecasting methods primarily focus on optimizing model architecture while neglecting the substantial impact of data quality on model learning. In this study, we aim to enhance model performance by optimizing data utilization based on data quality and propose a Data Quality-based Gradient Optimization (DQGO) method to facilitate training of recurrent neural networks. Firstly, we define sample quality as the matching degree between samples and model, and suggest using the attention entropy to calculate the sample quality through an attention mechanism. Secondly, we optimize the model's gradient vector by giving different weights to samples with different quality. Through experiments conducted on six datasets, the results demonstrate that DQGO significantly improves LSTM's performance. In certain cases, it even surpasses the state-of-the-art models.

KEYWORDS

Data quality,Gradient optimization,Time series, Recurrent neural network

1 INTRODUCTION

Time series prediction (TSF) is an indispensable artificial intelligence technology for various optimization control systems, such as the emergency traffic route planning system [1], power equipment intelligent maintenance system [2], and new energy generation plan [3]. However, existing TSF models based on Recurrent Neural Network (RNN) [4–6], Graph Neural Network (GNN) [7, 8], and Transformer framework [9–12] primarily focus on constructing robust models to facilitate temporal feature mining and achieve accurate predictions while overlooking the significant influence of data quality on model learning.

For time series, the discussion on the impact of noise data on model learning is necessary due to the common occurrence of low-quality (noise) time series data resulting from system failures or external interference in practical environments [13]. Although scholars have recently focused on investigating the influence of data on models [14], existing methods such as DataShapely [15], Influence function [16], and others [17–21] primarily address image and text data. However, these methods may not be optimal for analyzing time series data as they overlook its temporal dependence relationship.

Therefore, in this paper, we propose the **D**ata **Q**uality-based **G**radient **O**ptimization method to enhance the performance of RNN. In DQGO, our aim is to address two key issues: ❶ Quality evaluation for time series. We design a sample quality evaluation method based on attention mechanism to fully exploit temporal dependencies in sequences. ❷ Gradient optimization based on sample quality. We propose assigning different weights to samples according to their normalized quality scores, aiming to reducing the influence of low-quality samples.

To the best of our knowledge, this study represents the first attempt to evaluate data quality for time series through temporal dependence in sequences. Moreover, our proposed method allows for direct assessment of sample quality during model training, offering advantages over removal-based approaches. Experimental results on six datasets demonstrate that the LSTM model enhanced by DQGO outperforms current state-of-the-art models in terms of forecasting performance.

2 RELATED WORK

2.1 Time series forecasting

RNN is widely used in TSF, with LSTM and Gate Recurrent Unit (GRU) receiving extensive attention [6]. Transformer-based models are also a recommended solution for TSF. Informer [9], Autoformer [10], Pyraformer [22], FEDformer [11] and Crossformer [12] are representative models. Researchers often incorporate attention mechanisms and graph neural networks (GNNs) into model to improve predictive performance. For example, Guo et al. [7] designed a novel self-attention mechanism in GNN to predict traffic flow. Huang et al. [8] integrated the diffusion convolution neural network and a modified transformer to learn spatial-temporal dependence for traffic demand prediction. Existing works focus on re-structuring or optimizing neural network architecture, while overlook the influence of low-quality samples in practical time series data, which can significantly impact the representation learning of temporal dependence features.

*Corresponding author: Shan Li. This work is partially supported by the Sichuan Science and Technology Program (2023YFG0112); Intelligent Terminal Key Laboratory of Sichuan Province (Grant No.SCITLAB-20001);Postdoctoral research fund (2023SCU12093).

2.2 Sample quality evaluation

The sample quality-based methods for model training can be divided into two categories: removing-based and reordering-based. (1) Removing-based methods. These methods discard the so-called low-quality samples based on a valuation method [15, 16]. Mainstream approaches include data shapely [23], influence function [24], gradient-based influence function [16], and so on. These methods are associated with image classification tasks. Besides, some researchers propose to utilize sample feature to assess sample quality, such as accuracy, completeness, consistency and so on [25, 26]. For natural language data, the evaluation is based on simplicity and comprehensibility. (2) Reordering-based methods are known as the curriculum learning (CL) methods. Bengio mentioned in his original work that the basic idea of such methods is to first train the model with easy samples, and then gradually increase the difficult samples until the whole training datasets [27]. The automatic curriculum learning method which measures sample quality based on training loss is popular, with self-paced learning (SPL) and teacher-student learning (TSL) approaches receiving extensive attention [28, 29]. Removing-based methods directly delete samples, but it is difficult to determine how many samples should be deleted. Although reordering-based methods adjust the learning order of samples, the disadvantage of such methods is they ignore the cases where those hard-to-learn samples still affect model learning, especially in the later stages of the learning phase. Our proposed DQGO enables direct evaluation of sample quality during model training, providing advantages over removing-based and reordering-based methods.

3 PROBLEM FORMULATION

In this paper, denoted by $\boldsymbol{x}_i$ the i-th time series sample in dataset $\boldsymbol{X} \in R^{N\times(h+q)}$. N is the number of samples. The length of sample is $h+q$. The unbold letter x_i corresponds to an element in $\boldsymbol{x}_i$, and the superscript of x_i is used to indicate the time slot.

Problem 1. (**Time Series Forecasting**) *Given the sequence* $\boldsymbol{x}_i^{t-h+1:t} = [x_i^{t-h+1}, \ldots, x_i^t]$ *with length h, inferring* $\boldsymbol{x}_i^{t+1:t+q} = [x_i^{t+1}, \ldots, x_i^{t+q}]$ *the values in next q time slots based on a learnable model* $\mathcal{M}$*. Formally, the time series forecasting problem is defined as follows:*

$$\boldsymbol{x}_i^{t+1:t+q} = \mathcal{M}(\boldsymbol{x}_i^{t-h+1:t}) \tag{1}$$

Most existing studies on TSF assume that all samples have equal influence on optimizing $\mathcal{M}$. However, this assumption is overly idealistic. In this paper, we introduce the concept of sample quality to quantify the impact of each sample on the model.

Definition 1. (**Sample Quality**) *Given a sample* $\boldsymbol{x}_i$*, sample quality Q reflects how well sample* $\boldsymbol{x}_i$ *matches the model* $\mathcal{M}$*. Formally,*

$$Q = \mathcal{E}(\boldsymbol{x}_i|\mathcal{M}) \tag{2}$$

where $\mathcal{E}$ *is the metrics used to quantify sample quality.*

For instance, by considering LSTM as $\mathcal{M}$ and MAE as $\mathcal{E}$, Q is to evaluate how well $\boldsymbol{x}_i$ matches the LSTM based on MAE. Sample quality provides us the basis for how to optimize the impact of poorly matched samples (called as low-quality samples) on the model. Specifically, here MAE is merely provided as an illustrative example, we will define a better metrics $\mathcal{E}$ in next section. For simplicity, we use Q_i to denote the i-th sample's sample quality in the later chapters.

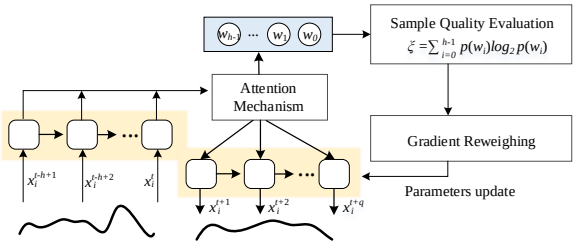


Figure 1: Framework of DQGO

4 OUR METHOD: DQGO

Figure 1 depicts the overview of DQGO. Firstly, we design an information entropy-based sample quality evaluation method which connects to the model through attention module. Secondly, we take the reweighted sample gradients as input and compute the update gradient vector for the model.

4.1 Sample Quality Evaluation

In this paper, we aim to enhance the prediction performance of RNN, so LSTM is selected as the basic model $\mathcal{M}$. For evaluating the metrics $\mathcal{E}$, we define a novel attention entropy (AE) which takes the attention weight as input.

This idea is motivated by the regression task. For example, $x_i^{t+1:t+q} = w_i^0 x_i^t + w_i^1 x_i^{t-1} + \cdots + w_i^{h-1} x_i^{t-h+1}$. w_i^τ $(\tau \in [0, h-1])$ is the item weight, which reflects the importance of its corresponding observation. For TSF task, we usually incorporate attention mechanism into LSTM, where the attention weights also serve as indicators of the significance of observed values. Furthermore, we posit that the attention distribution exhibits a correlation with the sample quality. These two characteristics can be modeled using information entropy based on the attention weight. To validate the correctness of our inference, we visually examined the correlation between AE and MAE using the ETTh1 dataset, as illustrated in Figure 2. The relationship between AE and MAE is inversely proportional, following a quadratic polynomial function ($AE = -0.6462 * MAE^2 - 0.1816 * MAE + 6.171$), exhibiting an R-square value of 0.8326 and an RMSE value of 0.0496. The results indicate that a sample with bigger AE will has smaller MAE and vice versa. In our previous study [6], we have validated that the model enhances its prediction accuracy by assigning greater importance to input data with similar shapes. Consequently, we propose the following physical explanation for the inverse relationship between them: each element in a high-quality sample should make a significant contribution to future values; conversely, only a few elements in a low-quality sample are crucial for predicting future values.

The calculation procedure of the proposed sample quality metric AE is presented as follows:

$$\mathbf{c}_i = [\mathbf{b}_i^\kappa \mathbf{e}_i^{0^T}, \ldots, \mathbf{b}_i^\kappa \mathbf{e}_i^{h-1^T}] \tag{3}$$

$$\mathbf{w}_i^\kappa = softmax(\mathbf{c}_i) \tag{4}$$

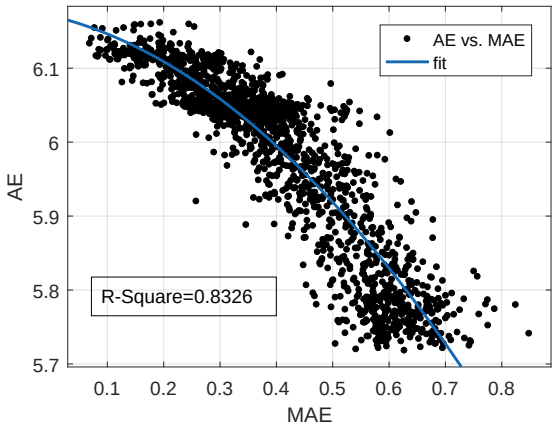


Figure 2: The relationship between AE and MAE on ETTh1 dataset

where $\mathbf{b}_i^{\kappa}$ is the hidden state of the κ-th decoder unit in LSTM and $\mathbf{e}_i^0, \ldots, \mathbf{e}_i^{h-1}$ are encoder hidden states. For q-steps prediction, sample i has the weight matrix $\mathbf{W}_i$:

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{w}_i^{t+1} \\ \vdots \\ \mathbf{w}_i^{\kappa} \\ \vdots \\ \mathbf{w}_i^{t+q} \end{bmatrix}_{q \times h} \tag{5}$$

By using column-wise sum operation, convert matrix $\mathbf{W}_i$ to vector $\boldsymbol{\omega}_i \in R^{1 \times h}$. Take the weights $\boldsymbol{\omega}_i$ as input, the sample quality can be calculated using the formula below.

$$\mathcal{E} = -\sum_{\tau=0}^{h-1} p(\omega_i^{\tau}) log_2 p(\omega_i^{\tau}) \tag{6}$$

In Equation (6), $p(\omega_i^{\tau}) = \frac{\omega_i^{\tau}}{\sum_{\tau=0}^{h-1} \omega_i^{\tau}}$ is the normalization of ω_i^{τ}. We define the metrics $\mathcal{E}$ as attention entropy (AE) based on the entropy of weight sequence ($[\omega_i^0, \ldots, \omega_i^{h-1}]$). As is mentioned above, when the value of AE is the greatest, we have $p(\omega_i^0) = p(\omega_i^1) = \cdots = p(\omega_i^{h-1}) = \frac{1}{h}$. This indicates that each element in the history sequence $\boldsymbol{x}_i^{t-h+1:t}$ is as important as $\boldsymbol{x}_i^{t+1:t+q}$. When the value of AE is lesser, it means that the attention weight is focused on a subset of the elements.

4.2 Sample Gradient Optimization

After obtaining the quality assessment, we designed a sample gradient optimization method. Specifically, based on the evaluation results of sample quality, we assign higher weights to high-quality samples and lower weights to low-quality ones, thereby mitigating the influence of low-quality samples on the model. Furthermore, this process can be seamlessly integrated into training without necessitating sample deletion or retraining.

The steps of sample gradient optimization are shown in Algorithm 1. In step 2, we normalize the sample quality and subsequently apply a reweighting factor to each sample's gradient, denoted as Q_i multiplied by the original gradient. In step 6, we take the mean of reweighting gradient vectors of samples as the optimal gradient update vector.

Algorithm 1 Sample Gradient Reweighing

Input:
Sample quality Q_i, $i = 1, \ldots, n$
Output:
Reweighing gradient $\boldsymbol{g}_i$
1: Initialize set $G \longleftarrow \{\emptyset\}$
2: Normalize Q_i using the max-min method
3: **for** each sample in a batch **do**
4: $\quad G \longleftarrow G \cup (Q_i * \boldsymbol{g}_i)$
5: **end for**
6: $G \longleftarrow \frac{1}{n} * \sum_{i=1}^{n} \boldsymbol{g}_i$, where $\boldsymbol{g}_i \in G$
7: Return G

5 EXPERIMENTS

In this section, we designed the following experimental approaches to evaluate the performance of DQGO:

- (1) Comparison with removing-based and reordering-based methods (in Section 5.2). Four representative removing-based methods are SGD-influence [24], g-shapely [23], GraNd [30] and VoG [31]. For the reordering-based methods, we adopt the Bootstrapping Curriculum Learning (BSCL) proposed in [32].
- (2) Comparison with the state-of-the-art TSF models (in Section 5.3). The LSTM, Informer [9], Crossformer [12], FEDformer [11], DLinear [33] and TimesNet [34] are used as baselines.
- (3) Adding noise to samples (in Section 5.4). Evaluate the efficacy of DQGO in handling noisy samples, comparing it with the methodologies proposed in Approach (1) and (2).

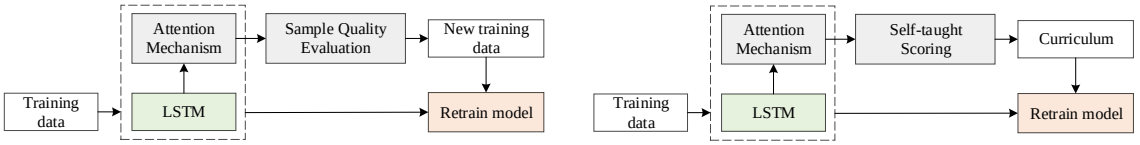
5.1 Experimental Setup

As shown in Figure 3(a), the removing-based methods including SGD-influence, g-shapely, GraNd and VoG, evaluate the sample quality first, and then retrain the model after removing some low-quality samples. BSCL is a reordering-based method (see Figure 3(b)). It reorders the training samples first using the proposed self-taught method, and then uses the reordering samples to train the model according to the curriculum plan and learning rate.

In experiments, the hidden size of an LSTM cell is set as 64, and the number of layers is 1. The settings of Informer, Crossformer, FEDformer, DLinear and TimesNet are consistent with the parameters in paper [34]. Unless otherwise specified, the task is to predict the next 96 values.

We evaluate our DQGO on six datasets including RED (Region Electricity Demand[1]), PeMS (Performance Measurement System[2]), two hourly ETT (Electricity Transformer Temperature[3]) datasets

[1]https://www.eia.gov/opendata/qb.php?category=3389943
[2]https://dot.ca.gov/programs/traffic-operations/mpr/pems-source
[3]https://github.com/zhouhaoyi/ETDataset



(a) SGD-influence, g-Shapely, GraNd and VoG (b) BSCL

Figure 3: Implementation details

ETTh1 and ETTh2, Electricity[4] and Exchange [35] dataset. To evaluate the prediction performance of each model, we use the mean absolute error (MAE), Mean Absolute Percentage Error (MAPE) and root mean square error (RMSE) as metrics.

5.2 Comparison with removing-based and reordering-based methods

In Table 1, DQGO has the obvious advantage on all of the six datasets except ETTh2 and Exchange. The results indicate that DQGO is more suitable for time series on data quality evaluation task. The quality assessment of time series in DQGO is realized through the lens of information entropy. Its primary advantage lies in its utilization of attention-based weights, which fully considers the interplay between historical series and predicted values, thereby better reflecting the data-model matching relationship. Additionally, the working principle based on information entropy enhances its interpretability, as demonstrated by the statistical results depicted in Fig. 2. Although methods such as g-shapely, SGD-influence, VoG, and GraNd have shown good performance on text and image data, their evaluation principle fails to capture the temporal dynamics in time series prediction by mining the matching relationship between data and prediction tasks. Furthermore, it has been observed from BSCL results that removing-based methods are not conducive to model learning. On the other hand, both BSCL and DQGO enhance model performance on the test set by increasing sample diversity without deleting any data. However, BSCL overlooks the presence of low-quality data which leads to suboptimal optimization effects for model learning compared to DGQO.

Table 1: Performance comparison for effectiveness validation

	Methods	DQGO	g-shapely	SGD-influence	BSCL	VoG	GraNd
RED	MAE	**0.1108**	0.1343	0.1252	0.1209	0.1269	0.1191
	RMSE	0.1611	0.1584	0.1662	0.1455	0.1522	**0.1444**
	MAPE	**18.42**	21.99	20.51	19.54	20.56	19.51
PeMS	MAE	**0.1678**	0.2071	0.2039	0.2037	0.2105	0.2017
	RMSE	**0.2310**	0.2579	0.2541	0.2527	0.2583	0.2483
	MAPE	**56.40**	101.35	101.94	98.86	108.40	104.19
ETTh1	MAE	**0.0509**	0.0537	0.0530	0.0517	0.0647	0.0576
	RMSE	0.0665	0.0694	0.0682	**0.0660**	0.0792	0.0713
	MAPE	**14.39**	15.37	15.27	14.89	18.63	16.59
ETTh2	MAE	0.0907	0.0866	0.0865	**0.0856**	0.0892	0.0899
	RMSE	0.1229	0.1103	0.1102	**0.1106**	0.1129	0.1131
	MAPE	14.78	14.05	14.04	**13.94**	14.55	14.78
Electricity	MAE	**0.0646**	0.0852	0.0813	0.0812	0.0799	0.0794
	RMSE	**0.0771**	0.1023	0.0994	0.1003	0.0973	0.0969
	MAPE	**12.25**	16.31	15.34	15.38	15.07	15.00
Exchange	MAE	0.0616	0.0455	0.0460	**0.0453**	0.0454	0.0459
	RMSE	0.0854	0.0575	0.0583	**0.0571**	0.0572	0.0579
	MAPE	8.74	6.49	6.58	**6.44**	6.45	6.55

[4]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

Table 2: Performance comparison with classical TSF models

	Methods	DQGO	LSTM	Informer	Crossformer	FEDformer	DLinear	TimesNet
RED	MAE	0.1108	0.5026	0.1523	0.1837	0.1365	**0.0972**	0.1117
	RMSE	0.1611	0.6757	0.1871	0.2277	0.1643	**0.1285**	0.1378
	MAPE	18.42	81.42	23.11	26.80	22.38	**14.54**	18.45
PeMS	MAE	0.1678	0.6662	0.2280	0.2146	0.3097	**0.1654**	0.2775
	RMSE	0.2310	0.8558	0.2982	0.2557	0.3728	**0.2099**	0.3294
	MAPE	**56.40**	346.72	102.67	118.25	190.21	77.18	176.53
ETTh1	MAE	**0.0509**	0.5643	0.1053	0.1162	0.1659	0.0748	0.0961
	RMSE	**0.0665**	0.7326	0.1314	0.1362	0.1776	0.0928	0.1155
	MAPE	**14.39**	157.70	31.10	33.46	47.21	23.74	29.52
ETTh2	MAE	0.0907	0.6122	0.1139	0.2323	0.1523	0.0921	**0.0865**
	RMSE	0.1229	0.7866	0.1554	0.2795	0.1742	0.1126	**0.1088**
	MAPE	14.78	101.02	18.06	34.99	24.93	14.65	**13.63**
Electricity	MAE	**0.0646**	0.6728	0.1121	0.1105	0.1309	0.0733	0.0774
	RMSE	**0.0771**	0.8833	0.1382	0.1266	0.1476	0.0895	0.0946
	MAPE	**12.25**	127.69	21.49	21.24	23.79	14.62	14.67
Exchange	MAE	**0.0616**	0.7043	0.3644	0.3358	0.1615	0.3042	0.0704
	RMSE	**0.0854**	0.8814	0.4480	0.3642	0.1758	0.3545	0.0874
	MAPE	**8.74**	96.79	49.07	42.98	22.38	37.40	10.07

Table 3: Impact of sample quality on DQGO

		noise_ratio=0		noise_ratio=0.3	
		lstm-att	DQGO	lstm-att	DQGO
powerLoad	MAE	↑0.1112	0.1108	↑0.1200	0.1106
	RMSE	↑0.1623	0.1611	↑0.1658	0.1548
	MAPE	↑0.1848	18.42%	↑19.98	18.43
PEMS04	MAE	↑0.1834	0.1678	↓0.1700	0.1938
	RMSE	↑0.2449	0.231	↓0.2243	0.2649
	MAPE	↑78.76	56.40	↑72.82	72.50
ETTh1	MAE	↑0.0581	0.0509	↓0.0622	0.0672
	RMSE	↑0.1159	0.0665	↓0.0915	0.1161
	MAPE	↑16.35	14.39	↓17.50	19.08
electricity	MAE	↑0.0651	0.0646	↑0.0890	0.0782
	RMSE	↑0.0792	0.0771	↑0.1458	0.1145
	MAPE	↑12.39	12.25	↑16.73	14.85
ETTh2	MAE	↓0.0795	0.0907	↑0.1035	0.0963
	RMSE	↓0.0987	0.1229	↑0.1495	0.1387
	MAPE	↓12.95	14.78	↑16.89	15.75
exchange	MAE	↓0.0561	0.0616	↑0.0817	0.0688
	RMSE	↓0.0729	0.0854	↑0.1426	0.0863
	MAPE	↓7.97	8.74	↑11.57	9.89

5.3 Comparison with the state-of-the-art TSF models

In the experiment, we conducted a comparative analysis between the DQGO-enhanced LSTM and the mainstream TSF model. Similar to Section 5.2, we employed "DQGO" as a representation of LSTM, and the experimental outcomes are presented in Table 2. Notably, DQGO exhibits superior performance on three datasets: ETTh1, electricity, and Exchange. Additionally, considering the potential involvement of the attention mechanism, we performed an ablation experiment comparing the performance of DQGO with that of attention-based LSTM (LSTM-ATT). In table 3, the results demonstrate that DQGO outperforms LSTM-ATT substantiating its effectiveness in enhancing LSTM's performance. Among SOTA models, both Dlinear and TimesNet exhibit better performance than Transformer-based models; specifically, Dlinear performs exceptionally well on RED and PeMS datasets while TimesNet excels on ETTh2 dataset. Overall findings from this experiment validate

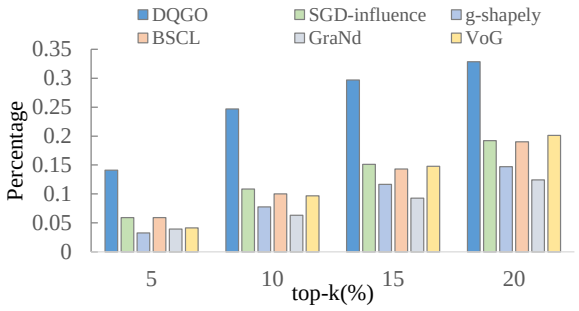


Figure 4: Identification of noisy samples

that incorporating DQGO can enhance LSTM's efficacy to achieve advanced levels.

5.4 Immunity to low-quality samples

In this section, we aim to test DQGO's immunity capacity to low-quality samples by adding noise to samples during training model. The ratio of samples with noise is set to 0.3. Firstly, we test its ability to identify the noise samples; Secondly, we compare its performance with data quality methods and SOTA prediction models.

Table 4: Impact of noise on the methods which have sample evaluation module

	Methods	DQGO	g-shapely	SGD-influence	BSCL	VoG	GraNd
RED	MAE	**0.1106**	0.1258	0.1331	0.1234	0.1237	0.1235
	RMSE	0.1548	0.1514	0.1806	0.1539	0.1491	**0.1489**
	MAPE	**18.43**	20.62	21.67	19.90	20.04	20.03
PeMS	MAE	**0.1938**	0.2135	0.2058	0.2060	0.2178	0.2301
	RMSE	0.2649	**0.2617**	0.2666	0.2681	0.2683	0.2852
	MAPE	**72.50**	110.78	103.53	100.92	119.63	135.32
ETTh1	MAE	0.0672	0.0536	0.0659	**0.0534**	0.0563	0.0565
	RMSE	0.1161	**0.0693**	0.0947	0.0696	0.0703	0.0818
	MAPE	19.08	**15.35**	19.19	15.37	16.16	16.68
ETTh2	MAE	0.0963	0.0881	0.0874	**0.0858**	0.0915	0.0868
	RMSE	0.1387	0.1108	0.1105	0.1111	0.1150	**0.1100**
	MAPE	15.75	14.43	14.23	**13.98**	15.04	14.19
Electricity	MAE	**0.0782**	0.0869	0.0819	0.0797	0.0818	0.0869
	RMSE	0.1145	0.1043	0.0996	**0.0972**	0.0993	0.1048
	MAPE	**14.85**	16.69	15.45	15.05	15.42	16.43
Exchange	MAE	0.0688	0.0483	**0.0450**	0.0454	0.0466	0.0579
	RMSE	0.0863	0.0612	**0.0568**	0.0571	0.0594	0.0700
	MAPE	9.89	6.92	**6.39**	6.45	6.64	8.22

Table 5: Impact of noise on TSF models

	Methods	DQGO	LSTM	Informer	Crossformer	FEDformer	DLinear	TimesNet
RED	MAE	0.1106	0.5696	0.1382	0.2042	0.1392	0.1501	**0.1073**
	RMSE	0.1548	0.7643	0.1733	0.2413	0.1672	0.1894	**0.1323**
	MAPE	18.43	94.61	22.52	29.60	22.91	21.83	**17.68**
PeMS	MAE	**0.1938**	0.7085	0.2145	0.2267	0.2836	0.2009	0.2770
	RMSE	0.2649	0.9094	0.2790	0.2679	0.3358	**0.2360**	0.3285
	MAPE	**72.50**	379.95	101.21	124.24	168.32	142.36	162.65
ETTh1	MAE	**0.0672**	0.7054	0.0903	0.1268	0.1591	0.1018	0.0678
	RMSE	0.1161	0.9133	0.1134	0.1566	0.1909	0.1234	**0.0840**
	MAPE	**19.08**	196.04	25.04	33.17	45.67	32.15	20.82
ETTh2	MAE	0.0963	0.6062	0.2377	0.2410	0.1305	0.1975	**0.0841**
	RMSE	0.1387	0.7861	0.2806	0.2731	0.1514	0.2354	**0.1048**
	MAPE	15.75	100.16	35.19	35.93	21.82	28.62	**13.64**
Electricity	MAE	**0.0782**	0.5169	0.1010	0.1222	0.1505	0.0987	0.0936
	RMSE	0.1145	0.5381	0.1201	0.1395	0.1668	0.1182	**0.1135**
	MAPE	**14.85**	98.29	20.16	23.54	29.39	19.87	17.83
Exchange	MAE	**0.0688**	0.6558	0.1878	0.3239	0.2311	0.5501	0.0701
	RMSE	**0.0863**	0.8369	0.2266	0.3532	0.2438	0.6109	0.0869
	MAPE	**9.89**	91.30	25.96	41.28	32.55	69.91	10.01

5.4.1 Identification of noisy samples. During training process, We added the noise sample manually. Then, we count the proportions of noise samples in the top 5%, 10%, 15% and 20% respectively. Fig. 4 shows the results on ETTh1 datasets. The results demonstrate that DQGO outperforms other methods in terms of identifying noisy samples. Moreover, from a DQGO perspective, noisy data can be identified based on the matching relationship between data and model since few data points noise sample can provide information for the prediction task.

5.4.2 Training with noisy samples. In this experiment, we will examine the performance of the DQGO and other algorithms when noise samples are included.

The prediction results of removing-based and reordering-based methods are presented in Table 4. It can be seen that DQGO outperforms other methods on RED, PeMS and Electricity datasets. g-shapely, SGD-influence, BSCL and GraNd all performed well on different data. These findings indicate that the inclusion of noise samples does impact the model by comparing Table 4 and Table 1; however, overall DQGO exhibits superior performance. In a similar way, we present the results of TSF models trained on noisy data in Table 5. It is evident that DQGO outperforms the state-of-the-art TSF models, except for RED and ETTh2 datasets; TimesNet demonstrates superior performance compared to Transformer-based models and DLinear. In Table 3, we also present the comparison of DQGO and attention-based LSTM in the case of training with noisy samples. The results indicate DQGO makes sense to reduce the influence of noisy samples on model learning.

We analyze the results from the following perspectives: (1) Based on the matching degree between the sample and the model, DQGO can filter the noise samples; (2) Comparing Table 5 with Table 4, it becomes apparent that deep learning models in Table 5 do not exhibit their advantages fully. These findings highlight that optimizing model learning from a data-quality based data utilization optimization strategy can yield favorable results even in the presence of noise.

6 CONCLUSION

In this paper, we have innovatively developed the Data quality-based gradient optimization method to facilitate the training of RNNs. Considering the occurrence of low-quality time series data is common due to system failures or external interference in practical applications, we aim to enhance the performance of TSF models by optimizing the utilization of data based on data quality. To this end, we initially developed a module for evaluating the quality of samples, which employs a novel metric known as attention entropy to quantify the quality of each sample. Subsequently, we proposed assigning higher weights to high-quality samples and lower weights to low-quality samples in order to optimize sample gradients and mitigate the impact of low-quality samples on model learning. Multiple experiments were conducted to validate the effectiveness of DQGO using an LSTM model. In the future, it is necessary to explore a more general quality assessment approach not only for RNNs. Additionally, for re-weighted gradients, we can also explore alternative solutions to solve gradients not only the mean operator used in DQGO.

REFERENCES

[1] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020.

[2] Weige Liang, Chi Li, Lei Zhao, Xiaojia Yan, and Shiyan Sun. Summarization of remaining life prediction methods for special power plants. *Applied Sciences*, 13(16):9365, 2023.

[3] Jelena Simeunović, Baptiste Schubnel, Pierre-Jean Alet, and Rafael E. Carrillo. Spatio-temporal graph neural networks for multi-site pv power forecasting. *IEEE Transactions on Sustainable Energy*, 13(2):1210–1220, 2022.

[4] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3882–3890. Curran Associates, Inc., 2016.

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

[6] Feihu Huang, Peiyu Yi, Jince Wang, Mengshi Li, and Jian Peng. Time-series forecasting with shape attention. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3299–3304. IEEE, 2022.

[7] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2022.

[8] Feihu Huang, Peiyu Yi, Jince Wang, Mengshi Li, Jian Peng, and Xi Xiong. A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences*, 594:286–304, 2022.

[9] 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, pages 11106–11115, 2021.

[10] 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, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430. Curran Associates, Inc., 2021.

[11] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR, 2022.

[12] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, pages 1–21, 2023.

[13] Jiaxian Chen, Ruyi Huang, Zhuyun Chen, Wentao Mao, and Weihua Li. Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective. *Mechanical Systems and Signal Processing*, 193:110239, 2023.

[14] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

[15] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning*, volume 97, pages 2242–2251. PMLR, 2019.

[16] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 4215–4224. Curran Associates, Inc., 2019.

[17] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In Luc De Raedt, editor, *Proceedings of the 31st International Joint Conference on Artifical Intelligence, IJCAI-ECAI 2022*, pages 5572–5579. International Joint Conferences on Artificial Intelligence Organization, July 2022.

[18] Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6388–6421. PMLR, 25–27 Apr 2023.

[19] Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. DAVINZ: Data valuation using deep neural networks at initialization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24150–24176. PMLR, 17–23 Jul 2022.

[20] Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6340–6347, 2020.

[21] Donghoon Lee, Hyunsin Park, Trung Pham, and Chang D Yoo. Learning augmentation network via influence functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10961–10970, 2020.

[22] 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*, pages 1–20, 2022.

[23] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8235–8243, 2021.

[24] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1885–1894. JMLR.org, 2017.

[25] Ikbal Taleb, Mohamed Adel Serhani, Chafik Bouhaddioui, and Rachida Dssouli. Big data quality framework: a holistic approach to continuous quality management. *Journal of Big Data*, 8(1):1–41, 2021.

[26] Sergey Redyuk, Zoi Kaoudi, Volker Markl, and Sebastian Schelter. Automating data quality validation for dynamic data ingestion. In *International Conference on Extending DB Technology*, pages 61–72, 2021.

[27] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[28] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022.

[29] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2694–2700, 2015.

[30] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20596–20607. Curran Associates, Inc., 2021.

[31] Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2022.

[32] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR, 09–15 Jun 2019.

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

[34] 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*, pages 1–23, 2023.

[35] 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*, SIGIR '18, page 95–104, New York, NY, USA, 2018. Association for Computing Machinery.