Towards an Efficient Algorithm for Time Series Forecasting with Anomalies

Hao Cheng [1] Qingsong Wen [2] Yang Liu [1] Liang Sun [2]

Abstract

Most of time series forecasting techniques assume that the training data is clean without anomalies. This assumption is unrealistic since the collected time series data can be contaminated in practice. The forecasting model will be inferior if it is directly trained by time series with anomalies. In this paper, we aim to develop methods to automatically learn a robust forecasting model from a data-centric perspective. Specifically, we first statistically define three types of anomalies in time series data, then theoretically and experimentally analyze the *loss robustness* and *sample robustness* when these anomalies exist. Based on our analyses, we propose a simple and efficient algorithm to learn a robust forecasting model which outperforms all existing approaches.

1. Introduction

Time series data collected inevitably has data bias, *i,e.,* they are contaminated by noises and anomalies (Wen et al., 2022). In time series forecasting with anomalies (TSFA), we study how to make robust and accurate forecasting on contaminated data. The earliest work of TSFA can date back to (Connor et al., 1994), where the classic **detection-imputation-retraining** pipeline is proposed. In this pipeline, a forecasting model is first trained using the data with outliers and noises. Then the trained model is used to predict values for each time step (**detection**). If the predicted value is far from the observed value, we will regard this time step as anomaly time step and then use the predicted value to replace the observed value on this time step (**imputation**). After imputation, the imputed (filtered) data are utilized to **retrain** a new forecasting model. The detection-imputation-retraining pipeline can work with any forecasting algorithm and improves performance in many TSFA tasks. However, it is very sensitive to the training threshold and anomaly types.

TSFA has very similar backgrounds to the research field of learning with noisy labels (LNL) (Natarajan et al., 2013; Song et al., 2022), but the settings are quite different. In LNL, the dataset consists of clean input samples X and the noisy labels $\widetilde{Y}$. For example, in the image classification task, some images may be wrongly annotated by humans. However, in the forecasting framework, the target in the past is often used as covariate to predict its future. Thus, in TSFA, the noise (anomaly) exists in both X and Y, making directly applying methods from LNL in TSFA challenging.

In this paper, we aim to build a unified framework for TSFA from a data-centric perspective. In particular, we have considered different types of anomalies in time series data, covering major anomaly types in many real-world applications. Based on the framework, we further propose a forecasting algorithm called RobustTSF, which is highly robust and outperforms SOTA in time series forecasting.

2. Preliminary and Formulation

Problem Formulation: Let $(z_1, z_2, \cdots, z_T)$ be a time series (without anomalies) of length T. Assume that the time series is partitioned into a set of time windows (with size α), where each window contains a sequence of predictors $\boldsymbol{x}_n = (z_{n-\alpha}, z_{n-\alpha+1}, \cdots, z_{n-1})$ and label $y_{\boldsymbol{x}_n} = z_n$, where $n \in \{1, 2, \cdots, T-\alpha\}$. This partition gives the training set $D = \{(\boldsymbol{x}_1, y_{\boldsymbol{x}_1}), \cdots, (\boldsymbol{x}_N, y_{\boldsymbol{x}_N})\}$ where $N = T - \alpha$, which can be assumed to be drawn according to an unknown distribution, $\mathcal{D}$, over $\mathcal{X} \times \mathcal{Y}$. In real-world applications, time series may have anomalies (anomaly types are defined later), and we can only observe time series with anomaly signals. The observed training set is denoted by $\widetilde{D} = \{(\widetilde{\boldsymbol{x}}_1, \widetilde{y}_{\boldsymbol{x}_1}), \cdots, (\widetilde{\boldsymbol{x}}_N, \widetilde{y}_{\boldsymbol{x}_N})\}$, where $\widetilde{o}$ denotes the observed value. The robust forecasting task aims to identify a model f that maps X to Y accurately using the observed training set $\widetilde{D}$ with anomalies.

Anomaly types: In the paper, we mainly consider point-wise anomalies. Define the noise (anomaly) rate as η where $0 < \eta < 1$. The observed value $\widetilde{z}_t$ in the time series can be represented as:

$$\widetilde{z}_t = \begin{cases} z_t & \text{with probability } 1-\eta \\ z_t^{\mathrm{A}}, & \text{with probability } \eta \end{cases}, \tag{1}$$

where z_t is the ground-true value and z_t^{A} is the value of anomaly. We consider three types of z_t^{A} as follows:

[1]University of California, Santa Cruz [2]DAMO Academy, Alibaba Group Bellevue, USA. Correspondence to: Qingsong Wen <qingsong.wen@alibaba-inc.com>.

- **Constant type anomaly**: $z_t^{\mathrm{A}} = z_t + \epsilon$, ϵ is constant.
- **Missing type anomaly**: $z_t^{\mathrm{A}} = \epsilon$, ϵ is constant.
- **Gaussian type anomaly**: $z_t^{\mathrm{A}} = z_t + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$.

$\mathcal{N}$ denotes Gaussian distribution, and the ϵ is noise scale.

3. Loss Robustness: Anomaly Effect in Y

Understanding how anomalies affect model performance is hard when anomalies exist in both X and Y. Thus we first assume the input X is clean and anomalies only exist in Y. In this case, $\widetilde{D} = \{(\boldsymbol{x}_1, \widetilde{y}_{\boldsymbol{x}_1}), \cdots, (\boldsymbol{x}_N, \widetilde{y}_{\boldsymbol{x}_N})\}$ (we can assume these data are non-overlapping segments of the time series). $\widetilde{y}_{\boldsymbol{x}_n}$ can be represented as:

$$\widetilde{y}_{\boldsymbol{x}_n} = \begin{cases} y_{\boldsymbol{x}_n} & \text{with probability } 1-\eta \\ y_{\boldsymbol{x}_n}^A, & \text{with probability } \eta \end{cases}.$$

Thus this setting is similar to the vanilla LNL setting except that for time series forecasting, the problem is regression instead of classification. Next, we examine the robustness of loss functions for different anomaly types defined earlier.

Theorem 3.1. *Let ℓ be the loss function and f be the forecasting model. Under Constant and Missing type anomalies with anomaly rate $\eta < 0.5$, if for each $\boldsymbol{x}$, $\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}) + \ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}^A) = C_{\boldsymbol{x}}$, where $C_{\boldsymbol{x}}$ is constant respect to the choice of f. Then we have:*

$$\mathbb{E}_{\widetilde{\mathcal{D}}}[\ell(f(X), \widetilde{Y})] = \gamma_1 \mathbb{E}_{\mathcal{D}}[\ell(f(X), Y)] + \gamma_2, \quad (2)$$

where $\gamma_1 > 0$ and γ_2 are constants respect to f.

Proposition 3.2. *From Theorem 3.1, MAE is robust to Constant and Missing type anomalies while MSE is not robust.*

Theorem 3.3. *Let ℓ be MAE or MSE loss function and f be the forecasting model. Under Gaussian type anomaly, let $f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\widetilde{\mathcal{D}}}[\ell(f(X), \widetilde{Y})]$. Then we have $f^*(X) = Y$.*

Theorems 3.1 and 3.3 show that MAE is trustworthy for the three types of anomalies defined earlier. Note that these conclusions are made statistically. If the size of the training set is relatively small or the anomaly rate is high, the performance may still drop. However, in our experiments, we find that MAE can be very robust in some real-world datasets.

4. Sample Robustness: Anomaly Effect in X

In this section, we analyze how anomalies in X affect the model performance, leaving the label clean. We first assume that *not all input samples contribute the same to the performance even though statistically, they have the same number of anomalies.* Our assumption implies that the position of anomaly matters. Experimentally, we find that when anomalies exist in the front or middle of input time series, the performance does not drop too much and is very close to the performance of clean data training. However, when anomalies exist in the back of time series, the performance significantly drops when the anomaly rate increases.

5. RobustTSF algorithm

By referring our analyses in Section 3 and Section 4, we show the procedure of our RobustTSF algorithm for time series forecasting with anomalies tasks.

Given a time series $\widetilde{\boldsymbol{z}} = \{\widetilde{z}_1, \cdots, \widetilde{z}_T\}$, we first calculate its trend $\boldsymbol{s} = \{s_1, \cdots, s_T\}$ by solving the optimization:

$$\min_{\boldsymbol{s}} \sum_{t=1}^{T} |\widetilde{z}_t - s_t| + \lambda \cdot \sum_{t=2}^{T-1} |s_{t-1} - 2 \cdot s_t + s_{t+1}|, \quad (3)$$

where λ is the hyper-parameter, s_t is the t-th value in $\boldsymbol{s}$. It is worth noting that Equation (3) is a variation of the original trend filtering algorithm (Kim et al., 2009). We change $(\widetilde{z}_t - s_t)^2$ to $|\widetilde{z}_t - s_t|$ to improve robustness. After trend filtering, we can segment $\boldsymbol{z}$ and $\boldsymbol{s}$ to get the training set $\widetilde{D} = \{(\widetilde{\boldsymbol{x}}_1, \boldsymbol{s}_1, \widetilde{y}_{\boldsymbol{x}_1}), \cdots, (\widetilde{\boldsymbol{x}}_N, \boldsymbol{s}_N, \widetilde{y}_{\boldsymbol{x}_N})\}$ which consists of N triplets. Note that $\boldsymbol{s}_n$ is the trend of $\widetilde{\boldsymbol{x}}_n$. Let $A(\widetilde{\boldsymbol{x}}_n)$ denote the anomaly score of $(\widetilde{\boldsymbol{x}}_n, \boldsymbol{s}_n, \widetilde{y}_{\boldsymbol{x}_n})$, which can be calculated as follows:

$$A(\widetilde{\boldsymbol{x}}_n) = \sum_{k=1}^{K} w(k) \cdot |\widetilde{\boldsymbol{x}}_n^k - \boldsymbol{s}_n^k|, \quad (4)$$

where K is the length of the input $\widetilde{\boldsymbol{x}}_n$, $\widetilde{\boldsymbol{x}}_n^k$ and $\boldsymbol{s}_n^k$ are the k-the value in $\widetilde{\boldsymbol{x}}_n$ and $\boldsymbol{s}_n$, respectively. $|\widetilde{\boldsymbol{x}}_n^k - \boldsymbol{s}_n^k|$ denotes the extent of anomaly at time step k. $w(k)$ is a weighting function that is designed non-decreasing. Specifically, we let $w(k) = 0$ when $k < K^{'}$ and $w(k) = 1$ when $k \geq K^{'}$.

The reason for designing $w(k)$ as a non-decreasing function is because the position of anomalies matters for the prediction performance, as we showed in Section 4. we enlarge the weights when anomalies are close to the labels. After calculating the anomaly scores of all the samples, we design the final loss in RobustTSF as:

$$L = \sum_{n=1}^{N} \mathbb{1}(A(\widetilde{\boldsymbol{x}}_n) < \tau) \cdot l(\widetilde{\boldsymbol{x}}_n, \widetilde{y}_{\boldsymbol{x}_n}), \quad (5)$$

where $\mathbb{1}()$ is the indicator function which is 1 when the condition is satisfied and 0 otherwise, τ is the threshold, and l is chosen to be MAE loss function. We term our method as RobustTSF. Note that RobustTSF is model agnostic and can use any DNN model (e.g., LSTM, Transformer, etc.) for time series forecasting.

We leave more discussion of RobustTSF and experiments into Appendix.

References

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

Bohlke-Schneider, M., Kapoor, S., and Januschowski, T. Resilient neural forecasting systems. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, pp. 1–5, 2020.

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

Connor, J. T., Martin, R. D., and Atlas, L. E. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

Forouzesh, M., Sedghi, H., and Thiran, P. Leveraging unlabeled data to track memorization. *arXiv preprint arXiv:2212.04461*, 2022.

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8527–8537, 2018.

Hao, C., Zhaowei, Z., Xing, S., and Liu, Y. Mitigating memorization of noisy labels via regularization between representations. *arXiv preprint arXiv:2110.09022*, 2022.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*, pp. 2304–2313, 2018.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. ℓ_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

Li, J., Socher, R., and Hoi, S. C. DivideMix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Li, W., Feng, C., Chen, T., and Zhu, J. Robust learning of deep time series anomaly detection models with contaminated training data. *arXiv preprint arXiv:2208.01841*, 2022.

Liu, Y. Understanding instance-level label noise: Disparate impacts and treatments. In *International Conference on Machine Learning*, pp. 6725–6735, 2021.

Liu, Y. and Guo, H. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning (ICML)*, pp. 6226–6236, 2020.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning (ICML)*, pp. 6543–6553, 2020.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204, 2013.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Wang, Y., Cai, Y., Liang, Y., Ding, H., Wang, C., Bhatia, S., and Hooi, B. Adaptive data augmentation on temporal graphs. *Advances in Neural Information Processing Systems*, 34:1440–1452, 2021a.

Wang, Y., Zhong, X., He, F., Chen, H., and Tao, D. Huber additive models for non-stationary time series analysis. In *International Conference on Learning Representations*, 2021b.

Wen, Q., Yang, L., Zhou, T., and Sun, L. Robust time series analysis and applications: An industrial perspective. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22)*, pp. 4836–4837, 2022.

Wu, Y., Ni, J., Cheng, W., Zong, B., Song, D., Chen, Z., Liu, Y., Zhang, X., Chen, H., and Davidson, S. B. Dynamic gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 651–659, 2021.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yoon, T., Park, Y., Ryu, E. K., and Wang, Y. Robust probabilistic time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 1336–1358. PMLR, 2022.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8778–8788, 2018.

Zhu, Z., Liu, T., and Liu, Y. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10113–10123, 2021a.

Zhu, Z., Song, Y., and Liu, Y. Clusterability as an alternative to anchor points when learning with noisy labels. In *International Conference on Machine Learning (ICML)*, pp. 12912–12923, 2021b.

The appendix is arranged as follows: in Section A, we prove the theorems in the main paper; in Section B, we deeply connect RobustTSF with LNL (Learning with Noisy Label) literature which builds a bridge between TSFA and LNL. in Section C, we provide experiments including experiments visualizing each anomaly type of time series, experiments verifying our claims in Section 3 and Section 4; experiments showing RobustTSF outperforms other SOTA methods.

A. Proof for Theorems

A.1. Proof for Theorem 3.1

Let $R_\ell(f) = \mathbb{E}_{\mathcal{D}}\ell(f(X), Y) = \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{x}}}\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}})$ and $R_\ell^\eta(f) = \mathbb{E}_{\widetilde{\mathcal{D}}}\ell(f(X), \widetilde{Y}) = \mathbb{E}_{\boldsymbol{x},\widetilde{y}_{\boldsymbol{x}}}\ell(f(\boldsymbol{x}), \widetilde{y}_{\boldsymbol{x}})$, respectively. Then

$$\begin{aligned}
&R_\ell^\eta(f)\\
=&\mathbb{E}_{\boldsymbol{x},\widetilde{y}_{\boldsymbol{x}}}\ell(f(\boldsymbol{x}), \widetilde{y}_{\boldsymbol{x}})\\
=&\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{y_{\boldsymbol{x}}|\boldsymbol{x}}\mathbb{E}_{\widetilde{y}_{\boldsymbol{x}}|\boldsymbol{x},y_{\boldsymbol{x}}}\ell(f(\boldsymbol{x}), \widetilde{y}_{\boldsymbol{x}})\\
=&\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{y_{\boldsymbol{x}}|\boldsymbol{x}}[(1-\eta)\cdot\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}) + \eta\cdot\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}^{\mathrm{A}})]\\
=&\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{y_{\boldsymbol{x}}|\boldsymbol{x}}[(1-\eta)\cdot\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}) + \eta\cdot(C_{\boldsymbol{x}} - \ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}))]\\
=&(1-2\cdot\eta)\cdot R_\ell(f) + \eta\cdot\mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{x}}}C_{\boldsymbol{x}}
\end{aligned}$$

where $\gamma_1 = 1 - 2\cdot\eta > 0$ and $\gamma_2 = \eta\cdot\mathbb{E}_{x,y_{\boldsymbol{x}}}C_{\boldsymbol{x}}$, which are constants respect to f.

A.2. Proof for Proposation 3.2

Let $q = f(\boldsymbol{x})$, then we are validating the value of $\ell(q, y_{\boldsymbol{x}}) + \ell(q, y_{\boldsymbol{x}}^{\mathrm{A}})$. For MAE loss, the value is represented as $|q - y_{\boldsymbol{x}}| + |q - y_{\boldsymbol{x}}^{\mathrm{A}}|$. From the property of absolute function, $|q - y_{\boldsymbol{x}}| + |q - y_{\boldsymbol{x}}^{\mathrm{A}}|$ achieves minimum when q lies in the range between $y_{\boldsymbol{x}}$ and $y_{\boldsymbol{x}}^{\mathrm{A}}$. Thus when $\min\{y_{\boldsymbol{x}}, y_{\boldsymbol{x}}^{\mathrm{A}}\} < q < \max\{y_{\boldsymbol{x}}, y_{\boldsymbol{x}}^{\mathrm{A}}\}$, we have $|q - y_{\boldsymbol{x}}| + |q - y_{\boldsymbol{x}}^{\mathrm{A}}| = C_{\boldsymbol{x}} = |y_{\boldsymbol{x}} - y_{\boldsymbol{x}}^{\mathrm{A}}|$.

This conclusion can not be fit for MSE loss, since $(q - y_{\boldsymbol{x}})^2 + (q - y_{\boldsymbol{x}}^{\mathrm{A}})^2$ is not constant when $\min\{y_{\boldsymbol{x}}, y_{\boldsymbol{x}}^{\mathrm{A}}\} < q < \max\{y_{\boldsymbol{x}}, y_{\boldsymbol{x}}^{\mathrm{A}}\}$.

A.3. Proof for Theorem 3.3

Following the notation in the proof for Theorem 3.1, we aim to show that $\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{y_{\boldsymbol{x}}|\boldsymbol{x}}[(1-\eta)\cdot\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}) + \eta\cdot\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}}^{\mathrm{A}})]$, $y_{\boldsymbol{x}}^{\mathrm{A}} \sim \mathcal{N}(y_{\boldsymbol{x}}, \sigma^2)$ achieves minimum when $f(\boldsymbol{x}) = y_{\boldsymbol{x}}$.

• For MSE loss, let $q = f(\boldsymbol{x})$, then we are to show that $(1-\eta)\cdot(q - y_{\boldsymbol{x}})^2 + \eta\cdot(q - y_{\boldsymbol{x}}^{\mathrm{A}})^2$ achieves minimum when $q = y_{\boldsymbol{x}}$. Since $(1-\eta)\cdot(q - y_{\boldsymbol{x}})^2$ achieves minimum when $q = y_{\boldsymbol{x}}$, we only need to prove $\int_{-\infty}^{+\infty} g(y_{\boldsymbol{x}}^{\mathrm{A}})\cdot(y_{\boldsymbol{x}}^{\mathrm{A}} - q)^2 dy_{\boldsymbol{x}}^{\mathrm{A}}$ achieves minimum when $q = y_{\boldsymbol{x}}$ where $g(y_{\boldsymbol{x}}^{\mathrm{A}}) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y_{\boldsymbol{x}}^{\mathrm{A}} - y_{\boldsymbol{x}})^2}{2\sigma^2})$ is the pdf of the distribution of $y_{\boldsymbol{x}}^{\mathrm{A}}$.

Let $s(q) = \mathbb{E}_{y_{\boldsymbol{x}}^{\mathrm{A}}}(y_{\boldsymbol{x}}^{\mathrm{A}} - q)^2 = \int_{-\infty}^{+\infty} g(y_{\boldsymbol{x}}^{\mathrm{A}})\cdot(y_{\boldsymbol{x}}^{\mathrm{A}} - q)^2 dy_{\boldsymbol{x}}^{\mathrm{A}}$, then

$$\begin{aligned}
&\frac{\partial s(q)}{\partial q} = 0\\
\Longrightarrow& 2\cdot q - 2\cdot\mathbb{E}_{y_{\boldsymbol{x}}^{\mathrm{A}}} g(y_{\boldsymbol{x}}^{\mathrm{A}}) = 0\\
\Longrightarrow& q = \mathbb{E}_{y_{\boldsymbol{x}}^{\mathrm{A}}} g(y_{\boldsymbol{x}}^{\mathrm{A}}) = y_{\boldsymbol{x}}
\end{aligned}$$

Thus $\int_{-\infty}^{+\infty} g(y_{\boldsymbol{x}}^{\mathrm{A}})\cdot(y_{\boldsymbol{x}}^{\mathrm{A}} - q)^2 dy_{\boldsymbol{x}}^{\mathrm{A}}$ achieves minimum when $q = y_{\boldsymbol{x}}$.

• MAE loss can also be proved with a similar procedure as follows.

Let $s(q) = \mathbb{E}_{y^{\mathrm{A}}_{\boldsymbol{x}}}|y^{\mathrm{A}}_{\boldsymbol{x}} - q| = \int_{-\infty}^{+\infty} g(y^{\mathrm{A}}_{\boldsymbol{x}}) \cdot |y^{\mathrm{A}}_{\boldsymbol{x}} - q| dy^{\mathrm{A}}_{\boldsymbol{x}}$, then

$$\begin{aligned}
&\frac{\partial s(q)}{\partial q} = 0 \\
\Longrightarrow &\mathbb{E}_{y^{\mathrm{A}}_{\boldsymbol{x}}}\left(\frac{-1 \cdot (y^{\mathrm{A}}_{\boldsymbol{x}} - q)}{|y^{\mathrm{A}}_{\boldsymbol{x}} - q|}\right) = 0 \\
\Longrightarrow &\mathbb{E}_{y^{\mathrm{A}}_{\boldsymbol{x}}}(\mathbb{1}\{y^{\mathrm{A}}_{\boldsymbol{x}} < q\} - \mathbb{1}\{y^{\mathrm{A}}_{\boldsymbol{x}} > q\}) = 0 \\
\Longrightarrow &\mathbb{P}(y^{\mathrm{A}}_{\boldsymbol{x}} < q) = \mathbb{P}(y^{\mathrm{A}}_{\boldsymbol{x}} > q) \\
\Longrightarrow &q = y_{\boldsymbol{x}}
\end{aligned}$$

Thus $\int_{-\infty}^{+\infty} g(y^{\mathrm{A}}_{\boldsymbol{x}}) \cdot |y^{\mathrm{A}}_{\boldsymbol{x}} - q| dy^{\mathrm{A}}_{\boldsymbol{x}}$ achieves minimum when $q = y_{\boldsymbol{x}}$.

B. Connection between RobustTSF and LNL

The goal of LNL is to mitigate the noisy label effect in classification and learn a robust model from the noisy dataset (Natarajan et al., 2013; Liu, 2021). The problem of LNL has been researched for a long time in the machine learning community, and many methods have been proposed in recent years, such as sample selection (Jiang et al., 2018; Han et al., 2018), robust loss design (Zhang & Sabuncu, 2018; Liu & Guo, 2020; Cheng et al., 2020; Zhu et al., 2021a), transition matrix estimation (Patrini et al., 2017; Zhu et al., 2021b), etc. Among all these methods, arguably the most efficient treatment is to adopt robust losses, since sample selection and noise transition matrix estimation always involve training multiple networks or need multi-stage training. We say a loss ℓ is **noise-robust** in LNL if it satisfies the following equation (Ghosh et al., 2017; Xu et al., 2019; Ma et al., 2020; Liu & Guo, 2020):

$$\arg\min_{f \in \mathcal{F}} \mathbb{E}_{(X,\widetilde{Y})}[\ell(f(X), \widetilde{Y})] = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(X,Y)}[\ell(f(X), Y)], \tag{6}$$

where f is the classifier, X is the input, Y and $\widetilde{Y}$ are clean labels and noisy labels, respectively. Equation (6) suggests that minimizing ℓ over clean dataset with respect to f is identical to minimizing ℓ over the noisy dataset. In this case, ℓ is robust to label noise. However, the robust losses in LNL cannot be directly applied to TSFA for two reasons: 1) Unlike LNL problem whose noisy labels are modeled in the classification setting, TSFA is dealing with regression problems; 2) Noise in LNL only exists in Y while anomalies in TSFA exist in both X and Y.

Next, we shown how RobustTSF connects to LNL:

- Denote the original observed inputs and labels as $\widetilde{X}$ and $\widetilde{Y}$. We first select $X^{'}$ from $\widetilde{X}$ where $X^{'}$ has little anomalies close to the labels. Denote the ground-truth label and observed label of $X^{'}$ as $Y^{'}$ and $\widetilde{Y'}$, respectively. From our analyses of sample robustness in Section 4, training on ($X^{'}$,$Y^{'}$) has very similar prediction performance compared to training on (X,Y). *I.e.,* $X^{'}$ acts like clean input time series. Mathematically, we have

$$\arg\min_{f} \mathbb{E}_{X,Y}[l(f(X), Y)] \approx \arg\min_{f} \mathbb{E}_{X^{'},Y^{'}}[l(f(X^{'}), Y^{'})].$$

- Since the selection process does not involve the labels, the distribution of $\widetilde{Y'}$ with respect to $X^{'}$ is unchanged. Let ℓ be the robust loss (MAE) in the setting of anomalies. From our analyses of loss robustness in Section 3, we have:

$$\arg\min_{f} \mathbb{E}_{X^{'},\widetilde{Y'}}[l(f(X^{'}), \widetilde{Y'})] \approx \arg\min_{f} \mathbb{E}_{X^{'},Y^{'}}[l(f(X^{'}), Y^{'})].$$

- Combining the above two equations, we have:

$$\arg\min_{f} \mathbb{E}_{X^{'},\widetilde{Y'}}[l(f(X^{'}), \widetilde{Y'})] \approx \arg\min_{f} \mathbb{E}_{X^{'},Y^{'}}[l(f(X^{'}), Y^{'})] \approx \arg\min_{f} \mathbb{E}_{X,Y}[l(f(X), Y)]. \tag{7}$$

The selection in Step 1 can refer to Section 5. Equation (7) implies that by using selective samples to train the model, the performance can be close to the clean samples training without anomalies in both inputs and labels. Equation (7) can be viewed as a generalization of Equation (6) which considers the noise in the inputs in TSFA tasks.

C. More Experiments

C.1. Visualizing each anomaly type

Figure 1 shows each anomaly type in the time series with different noise ratio.

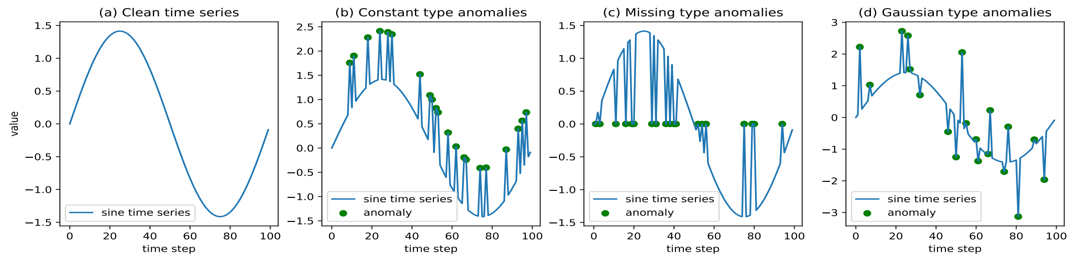

Figure 1. Visualization of time series with different types of anomalies. (a): Clean time series which is the sine function of time steps. We normalize the time series to 0 mean and 1 std. (b) (c) (d): Time series (sine) with Constant, Missing, and Gaussian type anomalies. The noise rate for these types of anomalies is 0.2. The noise scale is 1.0 for Constant and Gaussian anomaly and 0 for Missing anomaly.

Table 1. Comparison of loss functions on different anomaly types with anomaly rate 0.1 and 0.2. For each setting, we report the best performance on test set, presented as MAE/MSE. The model structure is LSTM. Detailed experimental setting is left in Appendix **??**.

Dataset	Loss	*Constant Anomaly*		*Missing Anomaly*		*Gaussian Anomaly*	
		$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.1$	$\eta = 0.2$
Electricity	MAE	0.185/0.070	0.187/0.071	0.189/0.071	0.195/0.078	0.184/0.070	0.186/0.073
	MSE	0.190/0.071	0.205/0.081	0.209/0.082	0.217/0.089	0.187/0.071	0.188/0.074
Traffic	MAE	0.194/0.108	0.199/0.112	0.206/0.119	0.210/0.123	0.198/0.112	0.206/0.117
	MSE	0.208/0.119	0.222/0.124	0.249/0.148	0.302/0.191	0.204/0.121	0.214/0.131

Table 2. Evaluating how different positions of anomalies in the input time series affect prediction performance. For each dataset, we simulate different types of anomalies with anomaly rates 0.1 and 0.2. The loss is MAE and the network structure is LSTM. Detailed setting is left in Appendix **??**.

Anomaly Type	Electricity with Anomaly Position			Traffic with Anomaly Position		
	front	middle	back	front	middle	back
Clean	0.182/0.070	0.182/0.070	0.182/0.070	0.194/0.105	0.194/0.105	0.194/0.105
Const. ($\eta = 0.1$)	0.182/0.070	0.183/0.071	0.195/0.078	0.196/0.105	0.200/0.115	0.215/0.131
Const. ($\eta = 0.2$)	0.182/0.069	0.183/0.071	0.219/0.094	0.194/0.109	0.195/0.106	0.236/0.153
Missing. ($\eta = 0.1$)	0.182/0/070	0.186/0.073	0.204/0.083	0.203/0.118	0.203/0.120	0.221/0.127
Missing. ($\eta = 0.2$)	0.182/0.070	0.183/0.071	0.223/0.105	0.204/0.116	0.205/0.116	0.251/0.160
Gaussian. ($\eta = 0.1$)	0.184/0.071	0.187/0.071	0.196/0.075	0.204/0.120	0.206/0.114	0.211/0.133
Gaussian. ($\eta = 0.2$)	0.184/0.171	0.186/0.072	0.213/0.087	0.206/0.121	0.212/0.127	0.226/0.147

C.2. Verifying our claims for loss robustness

we examine our Theorems in two real-world time series datasets: Electricity and Traffic, which are used by many time series forecasting papers (Yoon et al., 2022; Wang et al., 2021b; Wu et al., 2021). Since many works (Connor et al., 1994; Bohlke-Schneider et al., 2020; Li et al., 2022) dealing with TSFA problems deploy RNN-based structure to conduct experiments, we choose to use LSTM to validate our Theorems. Experiments are shown in Table 1. It can be observed that for Constant and Missing type anomalies, MAE is very robust whose performance does not vary too much when anomalies exist while MSE degrades the performance with the increase of the anomaly rate. For Gaussian-type anomalies, MAE and MSE can both be robust. These results support our theoretical analyses for loss robustness when anomalies only exist in the label.

C.3. Verifying our claims for sample robustness

To validate it, we simulate time series with anomalies located in different positions. A simple example with Gaussian type anomalies is shown in Figure 2. Note for Figure 2 (b) (c) (d), the number of anomalies is statistically the same. We perform experiments on Electricity and Traffic datasets to show how the position of anomalies affects the model performance in Table 2. It can be observed that when anomalies exist in the front or middle of input time series, the performance does not drop too much and is very close to the performance of clean data training. However, when anomalies exist in the back of time series, the performance significantly drops when the anomaly rate increases. Intuitively, it is understandable since the label is close to the back position of input time series. Interestingly, similar results can be found in temporal graph learning (Wang et al., 2021a) which shows that the more recent edges are more informative for given target predictions.

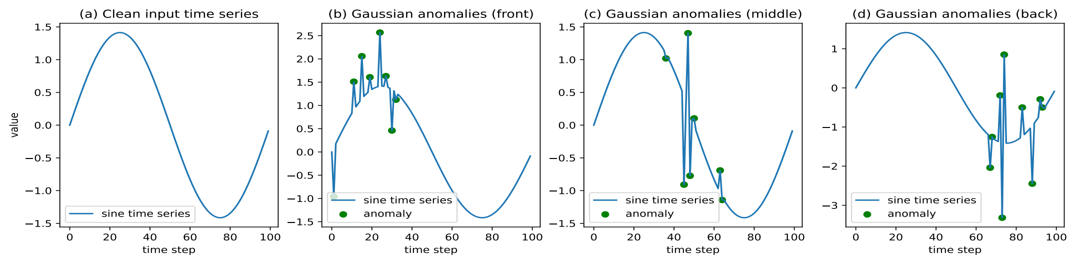

Figure 2. Visualization of time series with Gaussian anomalies in different positions. (a): Clean input time series which is the sine function of time steps. We normalize the time series to 0 mean and 1 std. (b) (c) (d): Time series (sine) with Gaussian anomalies located in front, middle and back of the input time series, respectively.

D. Comparing RobustTSF with SOTA methods

Here we demonstrate the effects of RobustTSF on TSFA tasks.

Model and Datasets Most of the recent papers on robust time series forecasting deploy LSTM-based structure to perform experiments (Yoon et al., 2022; Wang et al., 2021b; Wu et al., 2021). Additionally, all the related works on TSFA task also utilize LSTM. Thus we mainly adopt LSTM to perform experiments. Note all the methods use the same model structure for fair comparison. We use Electricity[1] and Traffic[2] datasets to evaluate each method. For each dataset, we split the dataset into training set and test set with a ratio of 7:3 and add anomalies in the training set. Note that we do not split a validation set from training set for TSFA task for two reasons: 1) since the training set is noisy, the validation set from the training set is not trustworthy for selecting models (Forouzesh et al., 2022) (We also demonstrate this in the Appendix); 2) Using the model selected from the validation set to report best performance in the test set is insufficient to evaluate the method for TSFA since a general consensus in LNL shows that DNN tends to fit clean samples and then fit noisy samples which cause the performance (accuracy) will gradually increase and then decrease (Arpit et al., 2017; Hao et al., 2022). Thus stability is also important for evaluating each method in TSFA task. Due to these two reasons, we follow the evaluation protocol from a very popular and benchmark method in LNL (Li et al., 2020) that for each training epoch, we record the performance of the DNN in the clean test set. After training ends, we report the **best** epoch test set performance and **last** epoch test set performance for each method.

Comparing methods and evaluation criterion: We consider the following methods for comparison: **Vanilla training**, which uses the model to directly train time series with MSE or MAE loss function, **Offline imputation (Connor et al., 1994)**, **Online imputation (Bohlke-Schneider et al., 2020)** and **Loss based sample selection (Li et al., 2022)**. We adopt MSE and MAE on test set to evaluate each method.

Training setup: Each method is trained for 30 epochs with ADAM optimizer. The learning rate is 0.01 for first 10 epochs and 0.001 for last 20 epochs. For RobustTSF, we fix $\lambda = 0.3$ (Equation (3)), $\tau = 0.3$ (Equation (5)), $K^{'} = K - 1$ (Dirac weighting) for all the experiments in the paper.

The experiments on single-step forecasting follows exact problem formulation in Section 2. The length of input sequence is 16 for Electricity and Traffic. The overall results are reported in Table 3 from which we can observe some interesting phenomenons:

- Among all the anomaly types, the missing anomaly has the largest negative effect on the model performance.
- Offline imputation, online imputation, and loss selection approaches do not have consistent improvement for all the anomaly types in each dataset. For example, online imputation method has improvements compared to vanilla training for missing anomaly type on the Traffic dataset but lower the performance for constant anomaly and Gaussian anomaly. The same phenomenon also happens to loss-based sample selection. The reason is unlike LNL, the noise (anomaly) in TSFA task exists in both X and Y, so small loss criterion in LNL does not fit well for TSFA tasks.
- Loss selection and RobustTSF are the most stable methods among all the methods (small Δ). They both select reliable samples for training but RobustTSF has better performance than loss selection approach.

[1]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams 20112014
[2]http://pems.dot.ca.gov

Table 3. Comparison of different methods on Electricity and Traffic dataset for single-step forecasting. We report the best and last epoch test performance for all methods, represented as MAE/MSE. We also report $\Delta = \overline{|\text{best} - \text{last}|}$, which is the average of $|\text{best} - \text{last}|$ for all anomaly settings of each method, to reflect the stableness of each method. The best results are highlighted in bold font.

Dataset	Method		*Clean*	*Constant Anomaly*		*Missing Anomaly*		*Gaussian Anomaly*		
			$\eta = 0$	$\eta = 0.1$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.3$	Δ
Electricity	Vanilla (MSE)	best	0.187/0.071	0.199/0.078	0.239/0.099	0.228/0.097	0.305/0.157	0.192/0.076	0.227/0.096	0.019
		last	0.205/0.082	0.219/0.091	0.258/0.117	0.229/0.098	0.354/0.197	0.213/0.087	0.255/0.124	
	Vanilla (MAE)	best	0.182/0.070	0.191/0.077	0.206/0.086	0.205/0.086	0.225/0.092	0.199/0.080	0.210/0.086	0.015
		last	0.198/0.081	0.223/0.096	0.245/0.109	0.215/0.090	0.245/0.107	0.203/0.082	0.219/0.096	
	Offline	best	0.184/0.071	0.187/0.071	0.214/0.087	0.190/0.074	0.212/0.084	0.184/0.070	0.193/0.075	0.009
		last	0.187/0.072	0.191/0.073	0.232/0.098	0.203/0.082	0.237/0.100	0.186/0.071	0.207/0.083	
	Online	best	0.183/0.070	0.199/0.077	0.225/0.091	0.195/0.075	0.227/0.093	0.194/0.074	0.209/0.084	0.01
		last	0.201/0.077	0.212/0.084	0.250/0.104	0.215/0.084	0.234/0.097	0.204/0.078	0.215/0.087	
	Loss sel	best	0.180/0.068	0.204/0.081	0.209/0.081	0.198/0.078	0.214/0.089	0.196/0.077	0.204/0.083	**0.004**
		last	0.182/0.069	0.208/0.085	0.217/0.089	0.205/0.083	0.217/0.090	0.205/0.083	0.210/0.086	
	RobustTSF	best	**0.177/0.065**	**0.177/0.066**	**0.183/0.070**	**0.181/0.068**	**0.194/0.074**	**0.176/0.067**	**0.177/0.067**	**0.004**
		last	**0.177/0.066**	**0.182/0.068**	**0.190/0.071**	**0.187/0.071**	**0.204/0.079**	**0.182/0.069**	**0.177/0.068**	
Traffic	Vanilla (MSE)	best	0.196/0.112	0.210/0.119	0.233/0.129	0.268/0.171	0.446/0.353	0.219/0.132	0.292/0.205	0.007
		last	0.199/0.116	0.220/0.126	0.235/0.130	0.275/0.173	0.451/0.357	0.233/0.144	0.304/0.226	
	Vanilla (MAE)	best	0.194/0.105	0.209/0.123	0.231/0.136	0.266/0.170	0.295/0.208	0.216/0.140	0.233/0.158	0.008
		last	0.202/0.114	0.210/0.123	0.245/0.151	0.272/0.186	0.307/0.220	0.224/0.148	0.234/0.159	
	Offline	best	0.189/0.104	0.208/0.123	0.222/0.133	0.221/0.135	0.280/0.186	0.205/0.120	0.218/0.125	0.005
		last	0.193/0.110	0.220/0.133	0.230/0.142	0.221/0.135	0.281/0.187	0.209/0.123	0.223/0.131	
	Online	best	0.206/0.118	0.216/0.113	0.239/0.126	0.220/0.116	**0.233/0.127**	0.224/0.127	0.244/0.141	0.016
		last	0.208/0.120	0.229/0.136	0.248/0.133	0.230/0.126	0.285/0.177	0.234/0.142	0.256/0.160	
	Loss sel	best	0.196/0.115	0.226/0.144	0.257/0.160	0.241/0.153	0.325/0.235	0.211/0.133	0.260/0.179	0.004
		last	0.199/0.120	0.232/0.151	0.260/0.170	0.241/0.153	0.329/0.238	0.220/0.138	0.260/0.179	
	RobustTSF	best	**0.185/0.100**	**0.198/0.113**	**0.203/0.111**	**0.200/0.113**	0.243/0.140	**0.185/0.101**	**0.202/0.116**	**0.003**
		last	**0.185/0.100**	**0.200/0.112**	**0.209/0.116**	**0.203/0.114**	**0.250/0.152**	**0.185/0.101**	**0.205/0.119**	

- RobustTSF has consistent improvements compared to vanilla training for all the anomaly types and all the datasets, which also suggests the usefulness of our selection module. It is very worth noting that even we do not manually add anomalies (clean), RobustTSF still has clear improvement. The reason is that the collected data recorded from sensors inevitably have anomalies. *Thus our method is also a general method for time series forecasting task.*