TARGETWEAVER: ADAPTIVE TARGET CONSTRUCTION FOR BACKDOOR ATTACKS ON MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Backdoor attacks on multivariate time series forecasting commonly assume predefined future targets, although suitable attack responses may vary across datasets, samples, and variables. We propose TargetWeaver, which adaptively selects target shape and temporal placement. TargetWeaver identifies dataset-relevant target patterns, selects context-dependent targets using temporal evidence and forecasting feedback, and trains pattern and offset selectors in successive phases. A historical trigger generator learns to induce the selected future responses. Across seven datasets and three forecasting architectures, TargetWeaver achieves the lowest average target-region error on six datasets while largely preserving clean forecasting accuracy. These results demonstrate the value of adaptive target construction alongside trigger optimization in forecasting backdoors.

1 INTRODUCTION

Multivariate time series forecasting supports traffic management, weather monitoring, energy operations, and computing-cluster scheduling. Recent architectures model these sequences through decomposition (Wu et al., 2021), frequency-domain representations (Zhou et al., 2022), and two-dimensional temporal variation (Wu et al., 2023). Their dependence on large historical datasets makes training-data integrity an important security concern (Li et al., 2024; Lin et al., 2024).

Backdoor attacks modify a small subset of training windows to associate historical triggers with future target responses while preserving normal forecasting behavior. BackTime learns historical triggers using temporal context and variable relationships, but uses predefined targets with constrained temporal placement (Lin et al., 2024). TDBA relaxes temporal alignment through variable-specific offsets and location-aware optimization, while retaining predefined target shapes (Liu et al., 2026). Existing forecasting backdoor attacks primarily rely on predefined, fixed target shapes.

Fixed target shapes may not suit datasets with different periodicity, fluctuations, and variable scales. Variation also arises within a dataset: different shape–offset combinations can yield different target-matching errors for the same input window. In an offline PEMS03 diagnostic, we hold the trained models fixed and enumerate six active patterns and six offsets. Each candidate configuration is applied to all selected variables, with errors measured separately for each sample–variable pair. The resulting errors vary across combinations, and the minimizing combination differs across sample–variable pairs (Figure 1). These observations motivate sample-dependent target construction, although target-matching error alone does not distinguish an easier target from a stronger trigger-induced response (Lin et al., 2024).

These observations raise the question of how to adapt target shapes and temporal locations to individual sample–variable pairs. Such adaptation requires accounting for local temporal context and the forecasting model's response to candidate targets, since target-matching error alone provides an incomplete basis for selection. Moreover, shape and location decisions are interdependent: the suitability of a temporal location depends on the selected shape and the corresponding candidate window. These requirements motivate a target-construction approach that uses contextual evidence and forecasting feedback to guide shape selection and shape-conditioned temporal placement.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

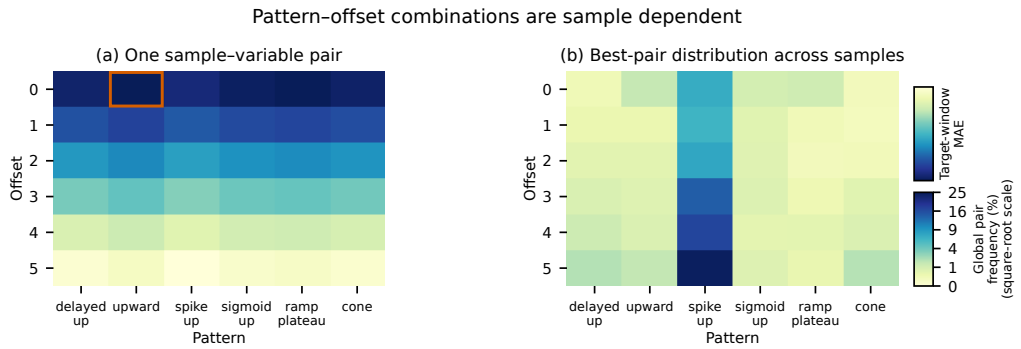


Figure 1: Evidence for sample-dependent pattern–offset selection on PEMS03. (a) Target-matching error for one representative sample–variable pair across six active patterns and six offsets; darker cells indicate lower target-window MAE, and the orange border marks the minimum. (b) Global distribution of the minimum-error pattern–offset pair across 256 sampled windows and 107 selected variables. The representative case in (a) was selected by the largest error range and is therefore not a typical sample. Each candidate is applied to all selected variables with trained models fixed. Both panels are offline diagnostics, not learned-selector frequencies or evidence that one combination is universally optimal.

To this end, TargetWeaver uses hierarchical candidate scoring and feedback-assisted decisions. A shared bank of 11 shapes is screened into a dataset-specific active set. PatternSelector scores patterns from historical context; OffsetSelector scores locations using pattern representations and offline future candidate evidence. A separate operation combines both scores with forecasting-model feedback to determine the target. An independent historical trigger generator is optimized to associate historical segments with the resulting future targets. The scoring modules are trained with a staged schedule described in the method section.

Our contributions are threefold. First, we formulate adaptive target construction for multivariate time series forecasting backdoors, treating target shape and temporal placement as decisions conditioned on dataset, sample, and variable characteristics. Second, we design TargetWeaver, a hierarchical target-construction framework that identifies dataset-relevant patterns and determines sample- and variable-specific targets through candidate scoring and forecasting-model feedback. Third, experiments across seven datasets and three forecasting architectures demonstrate the effectiveness of adaptive target construction, with the lowest average target-region error on six datasets while largely preserving clean forecasting accuracy. Ablations and sensitivity analyses further characterize the contributions and behavior of the proposed components.

2 Related Work

Multivariate forecasting. Temporal forecasting has been studied using recurrent models (Hochreiter & Schmidhuber, 1997; Lin et al., 2026), convolutional models (Luo & Wang, 2024), and graph-based approaches that capture dependencies among variables (Li et al., 2018; Wu et al., 2019). Transformer-based approaches (Vaswani et al., 2017; Zhou et al., 2021) combine attention with decomposition (Wu et al., 2021), frequency representations (Zhou et al., 2022), and stationarity modeling (Liu et al., 2022). PatchTST (Nie et al., 2023) and iTransformer (Liu et al., 2024) revisit tokenization, while linear models (Zeng et al., 2023) and TimesNet (Wu et al., 2023) provide alternative representations. TQNet (Lin et al., 2025) uses temporal queries to capture dependencies among variables, while TimeMixer (Wang et al., 2024a) mixes seasonal and trend information across multiple temporal scales using an MLP-based architecture. TimeXer incorporates exogenous variables through patch-wise self-attention and variate-wise cross-attention (Wang et al., 2024b). Other directions include diffusion-based generation (Yuan & Qiao, 2024), language-model reprogramming in Time-LLM (Jin et al., 2024), and cross-modality alignment in TimeCMA (Liu et al.,

2025). Large-scale pretraining for zero-shot forecasting is explored by TimesFM (Das et al., 2024) and the mixture-of-experts foundation model Time-MoE (Shi et al., 2025). These developments broaden the forecasting landscape; our transfer experiments concern FEDformer, Autoformer, and TimesNet, not pretrained foundation models.

Backdoors and forecasting attacks. Training-time backdoors associate a trigger with a designated model response, as demonstrated by BadNets (Gu et al., 2019). Broader threat models and defenses are reviewed by Li et al. (2024). Time series classification studies have explored generative trigger construction (Jiang et al., 2023) and frequency-domain injection (Huang et al., 2025). Unlike discrete classification labels, forecasting targets include continuous values, shapes, durations, and locations. BackTime associates historical triggers with future targets (Lin et al., 2024); TDBA introduces temporal decoupling and variable-specific placement (Liu et al., 2026). TargetWeaver builds on this setting by jointly adapting target shapes and offsets instead of prescribing a single shape.

Detection and mitigation. Backdoor defenses include representation analysis, trigger inversion, input screening, and model repair; a recent survey reviews these directions (Li et al., 2024). Recent approaches split potentially poisoned datasets to improve defense (Gao et al., 2023), unlearn shared adversarial examples (Wei et al., 2023), or purify training datasets using FLARE (Hou et al., 2025). For multivariate time series, USAD learns normal temporal behavior through adversarially trained autoencoders (Audibert et al., 2020). We use USAD to assess constructed historical windows, rather than treating one detector as evidence of robustness against all defenses.

3 Notation and Threat Model

3.1 Forecasting Task

For $\mathbf{X} \in \mathbb{R}^{T \times N}$, let T and N denote sequence length and variable count. At forecast origin t_i, the model receives $\mathbf{X}_i^h = \mathbf{X}[t_i - L_h : t_i, :]$ and timestamp features $\mathbf{M}_i$, and predicts

$$\widehat{\mathbf{X}}_i^f = f_{\boldsymbol{\theta}}(\mathbf{X}_i^h, \mathbf{M}_i), \qquad \mathbf{X}_i^f = \mathbf{X}[t_i : t_i + L_f, :]. \tag{1}$$

We distinguish the construction-stage surrogate f_q from the independently parameterized downstream model $f_{\boldsymbol{\theta}}$.

3.2 Threat Model and Objectives

Following BackTime and TDBA (Lin et al., 2024; Liu et al., 2026), the attacker can modify historical inputs and corresponding future labels in a small subset of training windows. The downstream forecasting architecture, training algorithm, and forecasting loss remain unchanged. In stage two, FEDformer, Autoformer, and TimesNet are trained from scratch on the constructed data. At activation, triggers are inserted into selected historical variables, and predictions are evaluated against the constructed future targets.

Construction uses a FEDformer surrogate. The offline protocol provides observed future windows and candidate forward outputs for feedback-assisted target decisions. Candidate evaluation uses only forward passes through the surrogate, while the trigger generator is trained by backpropagating gradients through the surrogate.

Attacked variables $\mathcal{S}$ and poisoning timestamps $\mathcal{T}$ are controlled by ratios α_s and α_t. Historical triggers have length L_g; future targets have length $L_p \leq L_f$ and must remain within the forecast horizon. The objectives are low target-region error on triggered inputs and retained forecasting performance on clean inputs. We additionally evaluate historical modification magnitude and detectability. Appendix A specifies the notation and selection budgets.

4 Method

Figure 2 summarizes stage-one learning. Target selection and historical trigger generation form separate branches that merge only when poisoned windows are constructed. Candidate feedback supervises the selectors; constructed windows support alternating surrogate and generator updates.

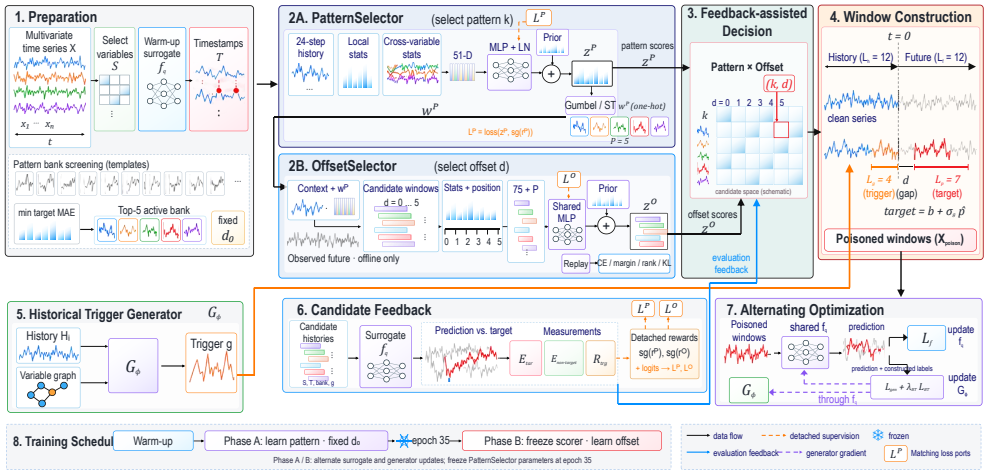


Figure 2: Overview of TargetWeaver stage-one training. (1) Preparation selects attacked variables and poisoning timestamps, warms up the surrogate, and screens an eleven-template bank to retain five active patterns. (2A–3) PatternSelector chooses a target pattern and OffsetSelector compares candidate locations using context, observed future windows, and positional evidence; their scores are combined in the feedback-assisted decision. (5) The independent historical trigger generator maps history and a learned variable graph to a trigger. (4) The selected future target and generated historical trigger are written into the same window to form a poisoned sample. (6) Candidate windows are evaluated by the surrogate to obtain target error, non-target error, and trigger-modification measurements; detached rewards supervise the two selectors. (7) The constructed windows drive alternating surrogate and generator updates, with generator gradients backpropagated through the shared surrogate. (8) Training warms up the surrogate, learns patterns at fixed offset d_0, and then freezes PatternSelector at epoch 35 while learning offsets.

4.1 Overview and Target Construction

We randomly select variables $\mathcal{S}$ and select high-error timestamps $\mathcal{T}$ using a warmed-up surrogate, subject to window and spacing constraints. For each selected sample i and variable s, construction determines a pattern and offset $d_{i,s} \in \mathcal{D} = \{0, \ldots, L_f - L_p\}$. Historical trigger $\mathbf{g}_{i,s}$ replaces the segment immediately before t_i. Future shape $\widehat{\mathbf{p}}_{i,s}$ is scaled by σ_g and anchored at $b_{i,s} = X[t_i - L_g - 1, s]$:

$$\begin{aligned} \widetilde{\mathbf{X}}[t_i - L_g : t_i, s] &= \mathbf{g}_{i,s}, \\ \widetilde{\mathbf{X}}[t_i + d_{i,s} : t_i + d_{i,s} + L_p, s] &= b_{i,s} + \sigma_g \widehat{\mathbf{p}}_{i,s}. \end{aligned} \tag{2}$$

The generator output is denormalized before replacing the historical segment. We use $L_h = L_f = 12$, $L_g = 4$, and $L_p = 7$, giving offsets 0–5.

4.2 Active Pattern Screening and Pattern Scoring

The universal bank contains 11 seven-step templates spanning gradual increases, steps, plateaus, and peaks of different widths and locations. A lightweight clean FEDformer evaluates pattern–offset combinations using simplified pattern-derived historical segments. Each pattern receives screening score $S_k = \min_{d \in \mathcal{D}} E^{\text{tar}}_{k,d}$, where $E^{\text{tar}}_{k,d}$ is validation target-region MAE. The five lowest-error patterns form the active bank $\widehat{\mathcal{P}}$.

PatternSelector scores active patterns for each sample and variable using 24-step historical context and local/cross-variable statistics. Its MLP uses layer normalization (Ni et al., 2024) and adds learnable global prior logits:

$$\mathbf{z}^p_{i,s} = F_p(\boldsymbol{\phi}^p_{i,s}) + \boldsymbol{\pi}^p, \qquad \widehat{\mathbf{p}}_{i,s} = \sum_{k=1}^{P} w^p_{i,s,k} \mathbf{p}_k. \tag{3}$$

Here $P = |\widehat{\mathcal{P}}|$, and weights $\mathbf{w}^p$ use Gumbel-Softmax with straight-through estimation (Jang et al., 2017). Balanced candidate sampling is annealed toward learned scores; the configured updates use the post-warm-up tail of this schedule. Feature dimensions, sampling rules, and pattern losses appear in Appendix B.3.

4.3 Candidate Offset Scoring and Feedback

OffsetSelector receives historical context, PatternSelector weights, and location-specific evidence. Each candidate includes its observed future window, window statistics, differences from historical values, and positional encoding. A shared scorer produces

$$z^o_{i,s,d} = F_o(\boldsymbol{\phi}^o_{i,s,d}) + \pi^o_d, \tag{4}$$

with learnable offset prior π^o_d. Temperature-scaled softmax and straight-through weights give discrete candidates. The shared scorer compares location-specific evidence across candidate offsets.

A decision step combines pattern scores, offset scores, and candidate forward evaluations. During learning, candidate quality provides detached rewards for each selector:

$$r^u_{i,s,a} = -E^{\mathrm{tar}}_{i,s,a} - \lambda^u_c E^{\mathrm{clean}}_{i,s,a} - \lambda^u_g R^{\mathrm{trg}}_{i,s,a}, \qquad u \in \{p, o\}. \tag{5}$$

Here a indexes patterns or offsets, and the terms measure target error, non-target error, and historical modification magnitude. Reward-derived losses update selectors even though rewards themselves stop gradients. Hard labels and margins apply only when the best-versus-second-best reward gap exceeds a threshold.

Offset learning fixes the current pattern and stores candidate features, labels, and reward vectors in a replay cache. Soft teacher distributions provide supervision in logit-based knowledge distillation (Sun et al., 2024). Here, soft distillation converts the complete reward vector into a teacher distribution, while pairwise ranking preserves relative preferences. Ranking applies to replay and separate pretraining batches, not the current candidate batch. Appendix B.4 gives the complete losses and gating rules.

4.4 Trigger Generation and Staged Optimization

Following BackTime (Lin et al., 2024), the historical trigger generator learns a sparse variable graph from frequency-domain features. Graph convolutions (Kipf & Welling, 2017) map six preceding historical steps to a four-step trigger, anchored to the latest observation and bounded in normalized space. Pattern and offset are not generator inputs; future targets affect it through the training loss.

The surrogate first warms up on clean windows. In phase A, we fix initial offset d_0, obtained by lightweight probing of up to three top-ranked active patterns. PatternSelector, the generator, and the surrogate are updated alternately. At epoch 35, phase B freezes PatternSelector and starts OffsetSelector learning while continuing alternating generator and surrogate updates. Generator/selector objectives are

$$\mathcal{L}^{(A)} = \mathcal{L}_{\mathrm{gen}} + \mathcal{L}_p + \lambda_{\mathrm{BT}}\mathcal{L}_{\mathrm{BT}}, \qquad \mathcal{L}^{(B)} = \mathcal{L}_{\mathrm{gen}} + \mathcal{L}_o + \lambda_{\mathrm{BT}}\mathcal{L}_{\mathrm{BT}}. \tag{6}$$

$\mathcal{L}_{\mathrm{gen}}$ combines temporally weighted prediction MSE over staggered windows with a variable-summed modification penalty. $\mathcal{L}_{\mathrm{BT}}$ penalizes time-summed changes; $\mathcal{L}_p$ and $\mathcal{L}_o$ supervise the respective selectors. The surrogate minimizes its own forecasting loss. Appendices B.5–B.6 specify these objectives and update steps.

After stage one, both selectors and the generator are frozen. Candidate scores and model-output feedback determine targets for final data construction, after which FEDformer, Autoformer, and TimesNet are trained from scratch. Each new dataset requires fresh statistics, screening, and component training (Appendix B.7).

5 Experiments

5.1 Experiment Setting

Data and models. Table 1 covers traffic (Song et al., 2020), weather (Max Planck Institute for Biogeochemistry, n.d.), transformer temperature/load (Zhou et al., 2021), and cluster CPU utilization (Alibaba, 2018). Each dataset is split chronologically into 60% training, 20% validation, and

20% testing, without shuffling. FEDformer serves as the stage-one surrogate. After the construction components are frozen, FEDformer, Autoformer, and TimesNet are trained from scratch on the constructed data for stage-two evaluation. The reported FEDformer results come from the stage-two model, which is separate from the stage-one surrogate. All forecasting input/output lengths are 12. Appendix C gives preprocessing and baseline details.

Table 1: Dataset statistics.

Dataset	Timestamps	Variables	Interval	Measurements
PEMS03	26,208	358	5 min	Traffic sensor flow
PEMS04	16,992	307	5 min	Traffic sensor flow
PEMS08	17,856	170	5 min	Traffic sensor flow
Weather	52,696	21	10 min	Weather observations
ETTh1	17,420	7	1 h	Transformer temperature and load
ETTm1	69,680	7	15 min	Transformer temperature and load
Alibaba Cluster 2018	2,304	256	5 min	CPU usage of 256 high-record-count machines

Compared methods. We compare three fixed trigger strategies—Random, Inverse, and Manhattan—together with the complete BackTime and TDBA methods (Lin et al., 2024; Liu et al., 2026). The three trigger strategies follow the corresponding definitions used in the BackTime evaluation, while BackTime and TDBA retain their method-specific target shapes, location policies, and losses.

Experimental settings. Where supported by the implementations, all methods use the same attacked-variable ratio, poisoning-timestamp ratio, input/output lengths, and forecasting architectures. The default settings are $\alpha_s = 0.30$, $\alpha_t = 0.03$, and normalized trigger budget $\epsilon = 0.20$. Both stages use 100 epochs and batch size 64. The forecasting-model and adaptive-component learning rates are 10^{-4} and 3×10^{-3}, respectively. PatternSelector is frozen at epoch 35, using zero-based epoch indices.

Metrics. Clean forecasting MAE (MAE_c) measures prediction error on unmodified test windows. For clean test-window set $\mathcal{Q}$,

$$\text{MAE}_c = \frac{1}{|\mathcal{Q}| L_f N} \sum_{i \in \mathcal{Q}} \left\| f_{\boldsymbol{\theta}}(\mathbf{X}_i^h, \mathbf{M}_i) - \mathbf{X}_i^f \right\|_1 . \tag{7}$$

Target-region MAE (MAE_a) measures how closely the trained model's predictions match each constructed target, evaluated only on the selected sample–variable pairs and target interval. Let $\mathcal{A}$ be those pairs, and let $d = d_{i,s}$ denote the selected offset for pair (i, s):

$$\text{MAE}_a = \frac{1}{|\mathcal{A}| L_p} \sum_{(i,s) \in \mathcal{A}} \left\| f_{\boldsymbol{\theta}}(\widetilde{\mathbf{X}}_i^h, \mathbf{M}_i)[d : d + L_p, s] - \widetilde{\mathbf{X}}_i^f[d : d + L_p, s] \right\|_1 . \tag{8}$$

Both metrics are computed after inverse normalization; lower is better. MAE_c is measured on clean test windows and is not the error outside the target interval of a constructed window. MAE_a is the mean absolute error between predictions and the target constructed by the corresponding method, evaluated within that target interval. For the Clean column, MAE_a is computed on unmodified inputs against a fixed 'cone' target at offset zero. Figure 3 gives example predictions around the trigger and target intervals; it illustrates target-region fitting alongside clean-input predictions, while MAE_c is aggregated separately over clean test windows. Historical modification magnitude is normalized by each variable's training-set standard deviation. USAD (Audibert et al., 2020) AUC-ROC measures separation between clean and constructed histories; 0.5 indicates chance-level detection and 1 perfect separation.

Figure 3 shows four PEMS03 sample–variable examples selected at the 25th, 50th, 75th, and 90th percentiles of diagnostic target MAE.

5.2 Main Results

Table 2 reports all model-level results and arithmetic averages. Bold marks the lowest MAEa per row. For each method, MAE_a is evaluated against the target constructed by that method. The Clean

column reports MAE_c on unmodified inputs and a reference MAE_a computed against a fixed ‘cone‘ target at offset zero, also on unmodified inputs.

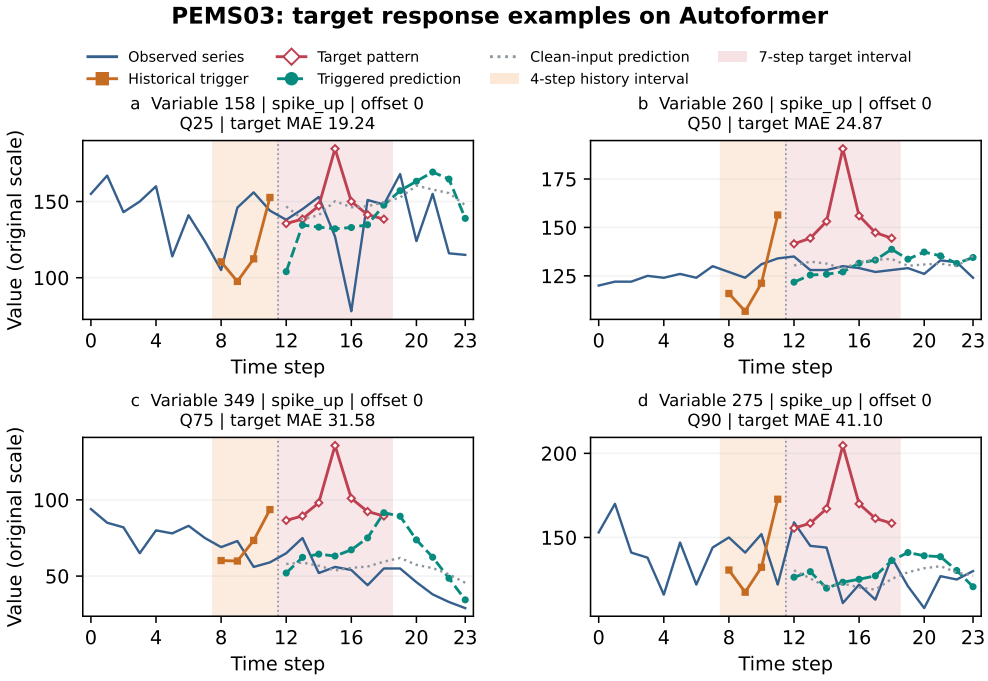


Figure 3: Autoformer target responses on PEMS03. Orange squares denote four-step historical triggers, and red diamonds denote seven-step target patterns. Green dashed and gray dotted lines show predictions from constructed and clean inputs, respectively. Light orange and light red backgrounds mark historical trigger and target intervals.

Table 2: Clean forecasting MAE and target-region MAE across seven datasets and three forecasting models. Lower is better.

Dataset	Model	Clean		Random		Inverse		Manhattan		BACKTIME		TDBA		Ours	
		MAEc	MAEa	MAEc	MAEa	MAEc	MAEa	MAEc	MAEa	MAEc	MAEa	MAEc	MAEa	MAEc	MAEa
PEMS03	TimesNet	20.75	41.91	22.43	28.78	22.62	23.33	22.60	35.08	21.82	20.87	21.44	23.67	20.48	**16.72**
	FEDformer	15.82	40.36	17.28	22.86	16.74	14.25	18.40	37.09	16.15	10.98	16.12	19.37	16.17	**8.56**
	Autoformer	16.46	38.47	17.95	26.90	17.58	18.44	17.90	39.34	17.44	16.58	17.26	20.60	16.85	**9.91**
	Average	17.68	40.25	19.22	26.18	18.98	18.67	19.63	37.17	18.47	16.14	18.27	21.21	17.83	**11.73**
PEMS04	TimesNet	23.92	52.73	27.80	35.60	25.93	31.59	27.15	34.93	25.10	30.31	25.29	42.28	24.68	**26.92**
	FEDformer	20.80	45.41	22.78	32.44	21.92	19.88	23.58	38.16	21.35	16.26	22.21	29.73	21.29	**12.97**
	Autoformer	21.15	45.49	23.56	39.85	22.92	22.98	22.45	41.34	22.51	23.99	23.03	31.65	22.08	**15.44**
	Average	21.96	47.88	24.71	35.96	23.59	24.82	24.39	38.14	22.99	23.52	23.51	34.55	22.68	**18.44**
PEMS08	TimesNet	22.78	45.13	27.00	31.30	24.43	29.46	27.77	34.77	25.16	25.98	24.28	31.19	25.40	**24.61**
	FEDformer	17.12	39.79	18.01	22.89	18.08	15.71	18.60	32.97	17.62	11.46	18.46	21.51	17.81	**10.09**
	Autoformer	18.23	38.66	19.24	28.41	17.95	19.80	19.90	42.96	18.94	17.86	18.86	22.98	18.47	**13.28**
	Average	19.38	41.19	21.42	27.53	20.15	21.66	22.09	36.90	20.57	18.43	20.53	25.23	20.56	**15.99**
Weather	TimesNet	23.40	90.54	27.30	41.13	21.81	34.25	27.48	46.43	32.82	32.03	41.12	80.20	15.33	**13.02**
	FEDformer	9.46	98.47	9.86	17.45	9.73	11.85	10.74	18.62	10.52	11.64	18.98	44.78	9.77	**7.54**
	Autoformer	10.88	93.75	10.20	18.10	11.59	23.32	10.38	19.32	9.06	9.59	8.65	29.29	8.93	**6.93**
	Average	14.58	94.25	15.79	25.56	14.38	23.14	16.20	28.12	17.47	17.75	22.92	51.42	11.34	**9.16**
ETTh1	TimesNet	2.03	5.24	2.28	3.05	2.11	**2.97**	2.33	3.86	2.13	3.14	2.08	4.10	2.22	3.03
	FEDformer	1.83	5.19	1.97	3.18	1.89	2.09	1.94	4.16	1.97	**1.84**	1.41	3.15	1.99	2.06
	Autoformer	1.84	5.19	1.93	3.61	1.96	2.16	2.00	3.50	2.06	**1.86**	1.85	3.90	2.05	1.89
	Average	1.90	5.21	2.06	3.28	1.99	2.41	2.09	3.84	2.05	**2.28**	1.78	3.72	2.09	2.33
ETTm1	TimesNet	1.34	2.46	1.56	2.33	1.38	1.60	1.58	2.63	1.34	1.52	1.39	1.93	1.35	**1.02**
	FEDformer	1.12	2.46	1.12	2.26	1.14	1.13	1.13	3.69	1.16	1.06	0.94	1.32	1.18	**0.60**
	Autoformer	1.30	2.58	1.28	2.14	1.25	1.09	1.24	2.30	1.17	1.01	1.16	1.55	1.15	**0.60**
	Average	1.25	2.50	1.32	2.24	1.26	1.27	1.32	2.87	1.22	1.20	1.16	1.60	1.23	**0.74**
Alibaba 2018	TimesNet	7.85	8.19	7.75	7.54	8.18	8.39	8.25	8.40	8.09	8.39	7.97	8.73	7.34	**6.55**
	FEDformer	7.36	5.92	6.92	6.45	6.93	4.63	7.21	7.39	6.76	5.37	6.03	6.02	5.97	**2.41**
	Autoformer	6.21	5.34	6.22	6.47	6.04	4.18	6.34	7.51	6.35	4.11	6.36	6.73	6.11	**2.83**
	Average	7.14	6.48	6.96	6.82	7.05	5.73	7.27	7.77	7.07	5.96	6.79	7.16	6.47	**3.93**

TargetWeaver achieved the lowest average MAEa on six datasets. Compared with BackTime, reductions were 4.41, 5.08, 2.44, 8.59, 0.46, and 2.03 on PEMS03, PEMS04, PEMS08, Weather, ETTm1, and Alibaba. ETTh1 was the exception: 2.33 versus BackTime's 2.28. TimesNet had higher target errors than FEDformer and Autoformer on all seven datasets, showing that target fitting depends on the forecasting architecture.

Compared with the Clean baseline, our average MAEc changed by +0.15 on PEMS03, +0.72 on PEMS04, +1.18 on PEMS08, -3.24 on Weather, +0.19 on ETTh1, -0.02 on ETTm1, and -0.67 on Alibaba. These modest changes, with MAEc decreasing on three datasets, indicate that our method largely preserves the forecasting models' clean-data performance rather than consistently degrading it.

5.3 Ablation Experiments

Table 3 evaluates stage-two Autoformer. Removing adaptive patterns replaces screening and PatternSelector with fixed `cone`. Removing adaptive offsets replaces OffsetSelector, candidate scoring, gating, replay, ranking, and distillation with random offsets 0–5. Removing staged training jointly trains all components without pattern freezing. Removing both adaptations uses fixed `cone` and random offsets. Other budgets and training settings remain unchanged.

Table 3: Component ablations on the stage-two Autoformer model. Target MAE is reported; lower is better.

Method	PEMS03	PEMS04	PEMS08	Weather	ETTh1	ETTm1	Alibaba
Full	**9.9117**	**15.4441**	**13.2839**	6.9306	**1.8852**	**0.5987**	2.8341
w/o Adaptive Pattern	13.4816	20.5333	17.2420	**6.2398**	1.9958	0.7451	3.4143
w/o Adaptive Offset	16.7599	22.1434	16.8364	33.8500	2.4126	0.9326	3.3573
w/o Staged Training	11.6190	16.9061	15.4606	9.5773	2.0934	0.6248	**2.7830**
w/o Pattern & Offset	20.8362	26.8915	20.9840	37.0183	2.7461	1.1926	3.9510

Removing adaptive offsets increased MAEa on all seven datasets, most sharply on Weather (6.9306 to 33.8500). Removing adaptive patterns increased MAEa on six datasets but decreased it on Weather (6.9306 to 6.2398). Removing both adaptations increased MAEa on all seven datasets.

Staged training yielded lower MAEa on PEMS03, PEMS04, PEMS08, Weather, ETTh1, and ETTm1, whereas joint training was slightly better on Alibaba, reaching 2.7830 versus Full's 2.8341. Appendix C.3 shows dataset-specific pattern preferences and a general preference for early offsets.

5.4 Parameter Sensitivity

Figure 4 varies one parameter at a time on PEMS03 with Autoformer. α_s controls the fraction of attacked variables, α_t controls poisoning timestamps, and ϵ bounds normalized generator output.

Increasing α_s from 0.10 to 0.30 reduced MAE_a from 14.12 to 9.91; increasing it to 0.50 raised MAE_a slightly to 10.01. Increasing α_t generally improved target fitting, although not monotonically. At 0.10, MAE_a reached 8.10 while MAE_c rose to 18.91. Increasing ϵ from 0.05 to 0.30 reduced MAE_a from 14.05 to 8.70, with MAE_c between 16.78 and 17.27.

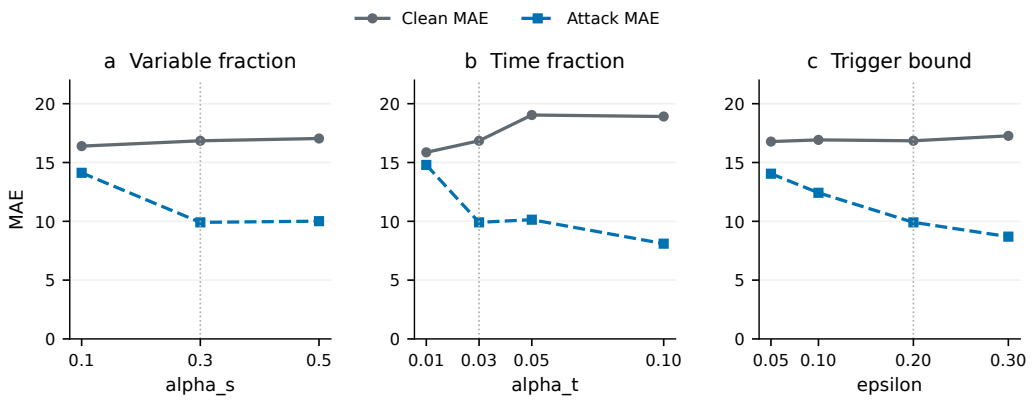


Figure 4: Parameter sensitivity on PEMS03 with Autoformer. Panels vary the attacked-variable ratio α_s, poisoning-timestamp ratio α_t, and historical trigger amplitude budget ϵ. Vertical dashed lines indicate the main-experiment defaults.

5.5 Historical Modification and Detectability

Table 4 reports normalized historical changes and USAD results. Magnitude statistics use 128 test windows. USAD uses 4,096 training windows and 128 test-window pairs, except Alibaba, which provides 1,359 training windows.

Table 4: Relative historical modification magnitudes and USAD anomaly detection results.

Dataset	Mean absolute change / variable SD	Median	95th percentile	USAD AUC-ROC
PEMS03	0.3318	0.2691	0.7378	0.5085
PEMS04	0.2804	0.2414	0.6101	0.5027
PEMS08	0.2494	0.1908	0.5775	0.5105
Weather	105.6442	1.5134	625.0097	1.0000
ETTh1	0.4896	0.2147	2.0095	0.4841
ETTm1	0.2784	0.1911	0.7127	0.5069
Alibaba 2018	6.2581	0.4445	1.3119	0.8243

AUC-ROC was near 0.5 on five datasets, but reached 1.0000 on Weather and 0.8243 on Alibaba. Weather had large normalized changes; a few near-constant Alibaba variables produced extreme outliers despite moderate median changes. Per-variable clipping increased target error in exploratory tests (Appendix C.4). These findings motivate adapting the generator to variable-specific scales while maintaining target fitting.

6 Conclusion

We revisited the use of predefined future targets in forecasting backdoor attacks. TargetWeaver makes target shape and temporal placement adaptive to individual samples and variables through hierarchical candidate selection, forecasting feedback, and staged optimization. Across seven multivariate datasets and three forecasting architectures, it achieved the lowest average target-region error on six datasets while largely preserving clean forecasting performance. Ablations show consistent gains from adaptive temporal placement, while the benefits of pattern adaptation and staged optimization vary across datasets. These findings highlight target construction as a consequential design choice in forecasting backdoors: the shape and placement of the desired response deserve consideration alongside the historical trigger used to induce it. The observed variation in historical modification magnitude and detectability across datasets motivates future work on scale-aware and continuity-constrained trigger generation.

AI USE STATEMENT

We used generative AI tools to assist with translation and with interpreting and articulating experimental results. We did not use generative AI tools for research methodology design, method implementation, or data generation or processing. Additionally, we used generative AI tools for manuscript polishing, scientific figure preparation, and literature search and reference organization. The authors take responsibility for the final content of this work, including all AI-assisted text and figures.

ETHICS STATEMENT

This work studies training-data backdoor attacks to characterize vulnerabilities in multivariate time-series forecasting systems. The experiments use research datasets and do not target deployed services.

REPRODUCIBILITY STATEMENT

The main text specifies the datasets, temporal split, forecasting horizons, poisoning ratios, model roles, training schedule, and evaluation metrics. The appendix records additional notation and implementation details.

REFERENCES

Alibaba. Alibaba cluster trace v2018. Public cluster trace dataset, 2018. URL https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2018. Accessed 2026-09-17.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3395–3404, 2020. doi: 10.1145/3394486.3403392. URL https://doi.org/10.1145/3394486.3403392.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4005–4014, 2023. doi: 10.1109/CVPR52729.2023.00390. URL https://doi.org/10.1109/CVPR52729.2023.00390.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068. URL https://doi.org/10.1109/ACCESS.2019.2909068.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Linshan Hou, Wei Luo, Zhongyun Hua, Songhua Chen, Leo Yu Zhang, and Yiming Li. FLARE: Toward universal dataset purification against backdoor attacks. *IEEE Transactions on Information Forensics and Security*, 20:6459–6473, 2025. doi: 10.1109/TIFS.2025.3581719. URL https://doi.org/10.1109/TIFS.2025.3581719.

Yuanmin Huang, Mi Zhang, Zhaoxiang Wang, Wenxuan Li, and Min Yang. Revisiting backdoor attacks on time series classification in the frequency domain. In *Proceedings of the ACM on Web Conference 2025*, pp. 1795–1810, 2025. doi: 10.1145/3696410.3714827. URL https://doi.org/10.1145/3696410.3714827.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Yujing Jiang, Xingjun Ma, Sarah Monazam Erfani, and James Bailey. Backdoor attacks on time series: A generative approach. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 392–403, 2023. doi: 10.1109/SaTML54575.2023.00034. URL https://doi.org/10.1109/SaTML54575.2023.00034.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, pp. 23857–23880, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2024. doi: 10.1109/TNNLS.2022.3182979. URL https://doi.org/10.1109/TNNLS.2022.3182979.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. SegRNN: Segment recurrent neural network for long-term time-series forecasting. *IEEE Internet of Things Journal*, 13(5):9861–9871, 2026. doi: 10.1109/JIOT.2025.3647705. URL https://doi.org/10.1109/JIOT.2025.3647705.

Xiao Lin, Zhining Liu, Dongqi Fu, Ruizhong Qiu, and Hanghang Tong. BackTime: Backdoor attacks on multivariate time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 131344–131368, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ed3cd2520148b577039adfade82a5566-Abstract-Conference.html.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. TimeCMA: Towards LLM-empowered multivariate time series forecasting via cross-modality alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):18780–18788, 2025. doi: 10.1609/aaai.v39i18.34067. URL https://doi.org/10.1609/aaai.v39i18.34067.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4054556fcaa934b0bf76da52cf4f92cb-Abstract-Conference.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, pp. 11116–11140, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Zhixin Liu, Xuanlin Liu, Sihan Xu, Yaqiong Qiao, Ying Zhang, and Xiangrui Cai. Beyond immediate activation: Temporally decoupled backdoor attacks on time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23999–24007, 2026. doi: 10.1609/aaai.v40i28.39577. URL https://ojs.aaai.org/index.php/AAAI/article/view/39577.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, pp. 31728–31770, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Max Planck Institute for Biogeochemistry. Weather data from the Jena weather station. Online dataset, n.d. URL https://www.bgc-jena.mpg.de/wetter/. Accessed 2026-09-17.

Yunhao Ni, Yuxin Guo, Junlong Jia, and Lei Huang. On the nonlinearity of layer normalization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37957–37998, 2024. URL https://proceedings.mlr.press/v235/ni24b.html.

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

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, pp. 34635–34667, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(1):914–921, 2020. doi: 10.1609/aaai.v34i01.5438. URL https://ojs.aaai.org/index.php/AAAI/article/view/5438.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15731–15740, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Sun_Logit_Standardization_in_Knowledge_Distillation_CVPR_2024_paper.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, pp. 38626–38652, 2024a. URL https://openreview.net/forum?id=7oLshfEIC2.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, pp. 469–498, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0113ef4642264adc2e6924a3cbbdf532-Abstract-Conference.html.

Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. In *Advances in Neural Information Processing Systems*, volume 36, pp. 25876–25909, 2023. doi: 10.52202/075280-1126. URL https://doi.org/10.52202/075280-1126.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019. URL https://www.ijcai.org/proceedings/2019/264.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, pp. 41582–41610, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325. URL https://ojs.aaai.org/index.php/AAAI/article/view/17325.

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*, pp. 27268–27286, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.