TOWARDS ROBUST TIME-SERIES FORECASTING: ATTACKS AND DEFENSE MECHANISMS

Linbo Liu [*]
Mathematics
UCSD

Youngsuk Park[†]
AWS AI Labs
Amazon

Trong Nghia Hoang
AWS AI Labs
Amazon

Hilaf Hasson
AWS AI Labs
Amazon

Jun Huan
AWS AI Labs
Amazon

ABSTRACT

This work studies the threats of adversarial attack on multivariate probabilistic forecasting models and viable defense mechanisms. Our studies discover a new attack pattern that negatively impact the forecasting of a target time series via making strategic, sparse (imperceptible) modifications to the past observations of a random (small) number of other time series. To mitigate the impact of such attack, we have developed two defense strategies. First, we extend a previously developed randomized smoothing technique in classification to multivariate forecasting scenarios. Second, we develop an adversarial training algorithm that learns to create adversarial examples and at the same time optimizes the forecasting model to improve its robustness against such adversarial simulation. Extensive experiments on real-world datasets confirm that our attack schemes are powerful and our defend algorithms are more effective compared with baseline defense mechanisms.

1 INTRODUCTION

Analyzing and improving prediction robustness for time-series forecasting models is a long-standing issue with broad applications in many disciplines such as climate change (Mudelsee, 2019), financial market analysis (Andersen et al., 2005), down-stream decision systems in retail (Böse et al., 2017), resource planning for cloud computing (Park et al., 2019), and optimal control of vehicles (Kim et al., 2020). The robustness issue originally is stemmed from the fact that time-series data often contain measurement noises and the statistical forecasting model can be very sensitive against such noises. Thus, developing forecasting models less sensitive to such noise while being able to preserve performance are highly desirable. Most of previous works (Liu & Zhang, 2021b; Wang & Tsay, 2021; Liu & Zhang, 2021a) in time series have therefore focused on (a) improving the robustness of many traditional, well-known statistical models such as vector auto-regressive and ARIMA (Brockwell & Davis, 2009), exponential smoothing (Brown, 1957) and Prophet (Taylor & Letham, 2018); or (b) improving model stability against outliers (Connor et al., 1994; Gelper et al., 2010). However, these approaches have not considered the possibility of adversarial noises which are strategically created to mislead the model rather than being distributed by a known distribution.

As a matter of fact, vulnerabilities against adversarial noises have been previously pointed out (Szegedy et al., 2013; Goodfellow et al., 2014b) in classification. For example, it has been demonstrated that human-imperceptible adversarial perturbation can alter classification outcomes of a deep neural net (DNN), revealing a severe threat to many safety-critical systems such as self-driving cars (Zhang et al., 2021). As such risk is often associated with the high capability to fit complex data pattern of DNN, it is possible that a similar threat can also occur in forecasting where traditional statistical models are being replaced increasingly by modern DNN-based forecasting models Salinas et al. (2019; 2020); Rangapuram et al. (2018); Lim et al. (2020); Wang et al. (2019); Park et al. (2022). For another example, imagine a situation in cost-critical financial market where a financial institute makes profits based on its prediction of its client's stock price, which might be attacked by adversaries who want to alter the financial institute's prediction. To make the attack hard to detect,

[*] Work done during the author's internship at AWS AI Labs.

[†] Correspondence to: Youngsuk Park pyoungsu@amazon.com .

the adversaries might devise a scheme that invests and hence changes the prices adversely for a small subset of stock indices, among a much larger pool of stock indices. Furthermore, the adversaries may not directly invest in the client's stock, which makes the attack even harder to detect as there is no direct adverse investment into the target stock.

However, such potential threats and simulation of a seemingly plausible and imperceptible attack to multivariate time-series forecasting (via an example of stock price prediction) are not straightforward to be formulated through the existing framework developed in classification settings. This is due to several setting differences between forecasting and classification, particularly in terms of unique characteristic of time series, e.g., time horizon, multiple items, and probabilistic predictions of forecasters, e.g., quantiles. To the best of our knowledge, although there have been several recent studies in this direction (Dang-Nhu et al., 2020; Yoon et al., 2022), they are all restricted to univariate forecasting where any via attacks must happen on the same target time series, which, unlike the threat in our stock prediction example above, are easier to detect.

Thus, it remains unclear under what settings and robustness requirements that adversarial perturbations can be substantiated into an attack for multivariate time-series forecasting; and whether it is defensible against such adversarial threats. Intuitively, under multivariate time-series scenarios, there are new regimes of sparse and indirect cross time series attack, which can be more dominant and effective than the direct attack substantiated in univariate case. Understanding whether such new regimes of attack exists and can be defended against is the main goal of our paper, which is achieved via addressing the following questions:

Indirectness. Can we mislead the forecasting of a time series via perturbations made to others?

Imperceptibility. Can the set of attacked time series be sparse and random to be less perceptible?

Defensibility. Can we defense against attacks with the above properties?

We address the above questions via the following technical contributions:

1. We devise a deterministic attack and show that adverse perturbations made to a subset of time series (not including the target time series) as described above can significantly alter the prediction outcome of the model. To be specific, we develop our deterministic attack (Section 3.2) based on the DeepVAR (Salinas et al., 2019) model, which currently provides state-of-the-art (SOTA) result to forecasting. Our attack is formulated as two-stage optimization task. The first phase finds an additive perturbation series to the authentic data such that DeepVAR's target statistics (e.g., prediction mean) is maximally altered in expectation, within a space of low-energy (hence, supposedly imperceptible) attacks. The second phase is then posed as a heuristic packing problem where all but k rows of the perturbation matrix are zeroed out such that minimal amount of attack effect is lost.

2. We develop probabilistic attack that learns to strategically make adverse perturbation to different (small) subsets of time series, making the attack much more stealth and harder to detect. Specifically, we formulate a probabilistic attack (Section 3.3) that relaxes the above k-hot constraint into a softer version that only requires the expectation of the attack vector rather than itself to be k-hot. Under such relaxation, we found that there is a provably approach to construct a learnable distribution over such k-hot attack space, with differentiable parameterization. This allows for the probabilistic attack model to elegantly merge the two separate phases of the deterministic attack. It can be shown empirically that an attack structured this way is often more effective (Section 5).

3. We propose two defense mechanisms. On the one hand, we adopt randomized smoothing technique (Cohen et al., 2019; Li et al., 2019) to our setting. On the other hand, we devise a defense mechanism (Section 4.2) based on the differentiable formulation of the probabilistic attack above. Our defense is generated as the optimal solution to a mini-max optimization task which minimizes the maximum expected damage caused by the probabilistic attacker that continually updates the generation of its adverse perturbation in response to the model updates. We also show the non-trivial effectiveness of our proposed defense against the aforementioned attacks (Section 5).

2 Related Work and Background

Deep Forecasting Models. The idea of applying neural network to time series forecasting dates back to Hu & Root (1964) and stayed relatively quiet for a few decades. Recently, with the growth

of large dataset and improvement of computing resources, more DNN-based forecasting models are investigated. Given the temporal dependency of time series data, RNN and CNN -based architectures have been proved a success for time series forecasting tasks (Rangapuram et al., 2018; Lim et al., 2020; Wang et al., 2019; Salinas et al., 2020) and Oord et al. (2016); Bai et al. (2018) respectively. In order to model the uncertainty, various probabilistic models have been proposed from distributional outputs Salinas et al. (2020); de Bézenac et al. (2020); Rangapuram et al. (2018) to distribution-free quantile-based outputs (Park et al., 2022; Gasthaus et al., 2019; Kan et al., 2022). In multivariate cases, Sen et al. (2019) leverages a global matrix factorization and a local temporal network . Salinas et al. (2019) generalize DeepAR (Salinas et al., 2020) to multivariate cases and employs low-rank Gaussian copula process to reduce problem complexity raised by high dimensionality. See Lim & Zohren (2021) for more comprehensive reviews.

Adversarial Attack. Despite its success in various tasks, deep neural network is especially vulnerable to adversarial attacks (Szegedy et al., 2013) in the sense that even imperceptible adversarial noise can lead to completely different prediction. In computer vision, many adversarial attack schemes have been proposed. See Goodfellow et al. (2014b); Madry et al. (2018) for attacking image classifiers and Dai et al. (2018) for attacking graph structured data. In the field of time series, there is much less literature and even so, most existing studies on adversarial robustness of MTS models (Mode & Hoque, 2020; Harford et al., 2020) are restricted to regression and classification settings . Alternatively, Yoon et al. (2022) studied both adversarial attacks to probabilistic forecasting models, which is only restricted to univariate settings.

Adversarial Robustness and Certification. Against adversarial attacks, an extensive body of work has been devoted to quantify model robustness and defense mechanisms accordingly, among which are Fast-Lin/Fast-Lip (Weng et al., 2018) recursively computing local Lipschitz constant of a neural network, PROVEN (Weng et al., 2019) certifying robustness in a probabilistic approach and DeepZ (Singh et al., 2018) based on abstract interpretation. To enhance model robustness, adversarial training and robust training are two popular techniques. In adversarial training (Madry et al., 2018; Wong et al., 2020), a neural network is trained on the adversarial examples instead of the original ones. As for robust training, one trains the model by simultaneously minimizing the loss and maximizing certified robustness (Weng et al., 2018; Wong & Kolter, 2018). Recently, randomized smoothing has gained increasing popularity as to enhance model robustness, which was proposed by Cohen et al. (2019); Li et al. (2019) as a defense approach with certification guarantee with several variants (Salman et al., 2019; Zhai et al., 2020; Kumar & Goldstein, 2021; Chiang et al., 2020) in the image classification setting. To the time series setting, Yoon et al. (2022) adopted randomized smoothing technique to univariate forecasting models and developed theory therein. However, to the best of our knowledge, there is no prior work applying randomized smoothing into multivariate probabilistic models.

3 Adversarial Attack Strategies

This section provides a quick review of the multivariate probabilistic forecasting model with associated adversarial framework (Section 3.1). Then, we introduce two class of sparse and indirect attacks. First, a deterministic approach is developed which optimizes for a deterministic set of time series to be altered adversely in order to attack a target time series. This is achieved via a two-stage optimization process (Section 3.2). Next, to equip the attack with uncertainty, a second non-deterministic approach is developed to instead optimize the attack effect for a distribution over such subset of time series, which (unlike the former approach) can be learned end-to-end in a single stage (Section 3.3).

3.1 Framework of Adversarial Attack against Multivariate Probabilistic Forecasting

Suppose a T-step history of a d-dimensional multivariate time series (MTS) $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \in \mathbb{R}^d$ are given. Let $x_{i,t} \in \mathbb{R}$ denote the observed value of i-th time series at time t. The forecasting task is to predict the values $\mathbf{x}_{T+1}, \mathbf{x}_{T+2}, \ldots, \mathbf{x}_{T+\tau}$ of the MTS τ-step into the future. The prediction is often based on the observed values of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \in \mathbb{R}^d$. A probabilistic forecasting model $p_\theta(\mathbf{z}|\mathbf{x})$ is often characterized as an auto-regressive function mapping from the observed input $\mathbf{x} \in \mathbb{R}^{d\times T}$ to a distribution over future target values $\mathbf{z} \in \mathbb{R}^{d\times\tau}$. Its parameterization θ is often associated

with a DNN in probabilistic deep forecasting models. Here, for notational convenience, we define $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{d\times T}$ and $\mathbf{z} = (\mathbf{x}_{T+1}, \mathbf{x}_{T+2} \ldots, \mathbf{x}_{T+\tau}) \in \mathbb{R}^{d\times\tau}$. Also, for a matrix A, let $A_{i,*}$ denote the i-th row of A. We define the element-wise maximum norm and Frobenius norm of A as $\|A\|_{\max} = \max_{i,j} |A_{ij}|$, $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$. For a specific form of θ, we refer interested readers to the DeepVAR paper (Salinas et al., 2019).

Sparse Adversarial Attack. We will now formally define an attack to the above forecasting model. In particular, suppose we are interested in the statistic $\chi(\mathbf{z}) \in \mathbb{R}^m$ that is a function of the random vector $\mathbf{z}$. To stage an attack, we let δ denote an adverse perturbation to the input $\mathbf{x}$ such that the Euclidean distance between the expected statistic $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x}+\delta)}[\chi(\mathbf{z})]$ and an adversarial target $\mathbf{t}(\mathbf{x}) \in \mathbb{R}^m$ is minimized. Here, $\mathbf{t}(\mathbf{x})$ is the desired target specified by the adversaries, which is radically different from the clean prediction $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\chi(\mathbf{z})]$. Thus, the optimal attack can be found via solving the following constrained minimization task:

$$\min_{\delta\in\mathbb{R}^{d\times T}} J(\delta) := \left\| \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x}+\delta)} [\chi(\mathbf{z})] - \mathbf{t}(\mathbf{x}) \right\|_2^2, \quad \text{s.t. } \|\delta\|_{\max} \leq \eta, \tag{3.1}$$

where η specifies the desired energy of the attack and the above expectation is over $\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{x}+\delta)$. Often, η is selected to be small to make the attack less perceptible. In the above, suppose that we want to mislead the forecasting for a set of pre-specified time-series with indices $I \in [d]$, our interested statistic will then be $\chi(\mathbf{z}) := \chi(\mathbf{z}_{I,*})$ which abstractly defines the predictive statistic corresponding to the rows of $\mathbf{z}$ with indices in I. Moreover, we can make the above attack sparse, hence stealthy. We now define a stealth attack δ as a sparse matrix such that its row sparsity as $s(\delta) = |\{i : \delta_{i,*} \neq \mathbf{0}\}| \leq k$ where k is the desired level of sparsity. Intuitively, this means a stealth attack is configured such that only a small subset of its row might be non-zero whereas the rest of it is zero. A small value of k would therefore make the attack even less perceptible. Furthermore, since the interested statistic $\chi(\mathbf{z}) = \chi(\mathbf{z}_{I,*})$ involves the indices of time series in I, setting $\delta_{I,*} = \mathbf{0}$ can make the attack more stealthy as there is no direct adverse alternation in the time series appearing in χ. Therefore, an optimal stealth attack can be found by adding these constraints to Eq. (3.1),

$$\min_{\delta\in\mathbb{R}^{d\times T}} J(\delta) := q\left\| \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x}+\delta)} [\chi(\mathbf{z})] - \mathbf{t}(\mathbf{x}) \right\|_2^2, \quad \text{s.t. } \|\delta\|_{\max} \leq \eta,\ s(\delta) \leq k,\ \delta_{I,*} = \mathbf{0}. \tag{3.2}$$

This however results in an intractable optimization task in general, so we provide two approximations in the subsequent sections.

3.2 Deterministic Sparse Attack

We first present a deterministic approach to solving (3.2) approximately. Here, the difficulty in optimizing (3.2) is due to the intractable constraint $s(\delta) \leq k$. To sidestep this, we use projected gradient descent (PGD) to numerically update the values of δ,

$$\delta^{(t+1)} = \prod_{B_\infty(0,\eta)} \left(\delta^{(t)} - \nabla_\delta J\big(\delta^{(t)}\big) \right), \tag{3.3}$$

where $\prod_{B_\infty(0,\eta)}$ is the projection onto the ℓ_∞-norm ball $B_\infty(0,\eta)$ with a radius η centered around the origin. Note that $\nabla_\delta J(\delta)$ involves the computation of the gradient of an expectation which is too complex to be analytically integrated. To overcome this intractability, we adopt the re-parameterized sampling approach used in Dang-Nhu et al. (2020) and Yoon et al. (2022). Suppose δ^* denote the converged value of δ following the iterative update in Eq. (3.3), we solve for its sparse approximation via

$$\hat{\delta} = \arg\min_{\delta} \quad \|\delta - \delta^*\|_F, \quad \text{s.t.} \quad s(\delta) \leq k, \quad \delta_{I,*} = \mathbf{0}. \tag{3.4}$$

It is straightforward to see that (3.4) can be solved analytically. Given δ^*, we compute the absolute row sum $c_i = \sum_{t=1}^T |\delta^*_{i,t}|$ for $i \in I^c$ and sort them in descending order $c_{\pi(1)} \geq \cdots \geq c_{\pi(d-1)}$. Rows from the top k index $\pi(1), \pi(2), \ldots, \pi(k)$ will be kept in $\hat{\delta}$ while the other will be zeroed out, as described in Algorithm 1.

3.3 Probabilistic Sparse Attack

In this subsection, we further remove the two-stage heuristic approximation in Section 3.2, which is non-differentiable and cannot be optimized end-to-end, making it unsuitable to be integrated into a

differentiable defense mechanism as described later in Section 4.2. The key issues here are in fact the non-convex and non-differentiable constraint in (3.2) which disables differentiable optimization via gradient descent. To sidestep this, we instead view the sparse attack vector as a random vector drawn from a distribution with differentiable parameterization which can be learned via gradient updates.

The core challenge here is how to configure such distribution whose support is guaranteed to be within the space of sparse vectors. To achieve this, we configure this distribution as a learnable combination of a normal standard and a Dirac density, whose samples can be interpreted as differentiable transformation of samples drawn from a parameter-free normal standard – see Theorem 3.1. As we can update the parameters of the transformation via its gradient, we can learn the attack distribution with sparse support – see Theorem 3.2. Key to this parameterization is the ability to sample from a combination between Dirac and Gaussian densities, which is substantiated via the construction of a sparse layer as detailed below.

Sparse Layer. A sparse layer is configured as a conditional distribution $q_\Theta(\delta|\mathbf{x})$ such that $\mathbb{E}[s(\delta)] \leq k$ and $\delta_{I,*} = \mathbf{0}$ where $\delta \sim q_\Theta(\delta|\mathbf{x})$, I is the set of target time-series which are not to be altered, and k is the user-specified level of sparsity as defined before. Let $\Theta = (\beta, \gamma)$. We treat each row $\delta_{i,*}$ of δ as an independent sample drawn from $q_i(\delta_{i,*}|\mathbf{x}; \beta, \gamma)$ parameterized by β and γ, as defined below:

$$q_i\Big(\delta_{i,*}|\mathbf{x}; \beta, \gamma\Big) := r_i\left(\gamma\right) \cdot q'_i\Big(\delta_{i,*}|\mathbf{x}; \beta\Big) + \Big(1 - r_i(\gamma)\Big) \cdot D\Big(\delta_{i,*}\Big), \tag{3.5}$$

where $r_i(\gamma) := (k\gamma_i^{1/2}/\sqrt{d})/(\sum_{i=1}^d \gamma_i)^{1/2}$ are the combination weights, $D(\delta_{i,*})$ is the Dirac density concentrated at $\delta_{i,*} = \mathbf{0}$ and $q'_i(\delta_{i,*}|\mathbf{x}; \beta)$ is a Gaussian density whose mean and variance are functions of $\mathbf{x}$ which are parameterized by β that can be weights of a DNN. The combination weight $r_i(\gamma)$ on the other hand denotes the probability mass of the event $\delta_{i,*} = \mathbf{0}$, which is parameterized by γ. Intuitively, this means the choice of $\{r_i(\gamma)\}_{i=1}^n$ controls the row sparsity of the random matrix δ, which can be calibrated to enforce that $\mathbb{E}[s(\delta)] \leq k$. We will show in Theorem 3.1 how samples can be drawn from the combined density in (3.5). Then, Theorem 3.2 to show why sample δ drawn from (3.5) would meet the constraint $\mathbb{E}[s(\delta)] \leq k$. Put together, Theorem 3.1 and Theorem 3.2 enables differentiable optimization of a sparse attack distribution as desired.

Lemma 3.1. *Let* $\delta'_{i,*} \sim q'_i(\cdot|\mathbf{x}, \beta)$ *and* $u_i \sim \mathcal{N}(0, 1)$ *for* $i = 1, \ldots, d$. *Define* $\delta_{i,*} = \delta'_{i,*} * \mathbb{I}(u_i \leq \Phi^{-1}(r_i(\gamma)))$. *Then,* $\delta_{i,*} \sim q_i(\delta_{i,*}|\mathbf{x}; \beta, \gamma)$.

Here, $q_i(\cdot|\mathbf{x}; \beta, \gamma)$ is given in (3.5) and Φ^{-1} is the inverse cumulative of the standard normal distribution. We provide the proof in the appendix.

For implementation, observing that the second property $\delta_{I,*} = \mathbf{0}$ can always be satisfied by zeroing out the I rows of δ. Thus, for simplicity, we ignore this constraint. Let $q'_i(\cdot|\mathbf{x}; \beta)$ be dense distributions, e.g. $\mathcal{N}(\mu(\beta), \sigma^2(\beta)I)$, over $\mathbb{R}^T$ and $u_i \sim \mathcal{N}(0, 1)$ for $i \in [d]$. We can construct a binary mask as $\text{mask}_i = \mathbb{I}(u_i \leq \Phi^{-1}(r_i(\gamma)))$, $i \in [d]$, where $r_i(\gamma) = (k\gamma_i^{1/2}/\sqrt{d})/(\sum_{i=1}^d \gamma_i)^{1/2}$.

Next, for each $i \in [d]$, we draw $\delta'_{i,*}$ from $q'_i(\cdot|\mathbf{x}, \beta)$ and obtain $\delta_{i,*}$ by $\delta_{i,*} = \delta'_{i,*} * \text{mask}_i$, where $*$ is element-wise multiplication . Finally, we set $\delta_{I,*} = \mathbf{0}$. Theorem 3.2 below then verifies the required sparsity property in expectation, thus completing our differentiable sparse attack.

Lemma 3.2. *Let* $\delta \sim q_\Theta(\cdot|\mathbf{x})$. *Then,* $\mathbb{E}[s(\delta)] \leq k$.

We provide the proof in the appendix.

Optimizing Sparse Layer. The differentiable parameterization of the above sparse layer can be optimized (for maximum attack impact) via minimizing the expected distance between the attacked statistic and adversarial target:

$$\min_{\Theta} \;\; \mathbb{E}_{\delta \sim q_\Theta(.|\mathbf{x})} \Big\| \mathbb{E}_{\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{x}+\delta)}[\chi(\mathbf{z})] - \mathbf{t}(\mathbf{x}) \Big\|_2^2, \tag{3.6}$$

This attack is probabilistic in two ways: First, the magnitude of the perturbation δ is a random variable from distribution $q(\cdot|\mathbf{x})$. Second, the non-zero components of the mask depend on the

random Gaussian samples, which brings another degree of non-determinism into the design, making the attack more stealth and harder to detect.

Discussion. There are three important advantages of the above probabilistic sparse attack. First, by viewing the attack vector as random variable drawn from a learnable distribution instead of fixed parameter to be optimized, we are able to avoid solving the NP-hard problem (3.2) as usually approached in previous literature (Croce & Hein, 2019). Second, our approach introduces multiple degree of non-determinism to the attack vector, apparently making it more stealth and powerful (see the experiments in Section 5). Last, as the attack model is entirely differentiable, it can be directly integrated as part of a differentiable defense mechanism that can be optimized via gradient descent in an end-to-end fashion – see Section 4.2 for more details.

4 Defense Mechanism against Adversarial Attacks

The adversarial attack on probabilistic forecasting models was investigated in Dang-Nhu et al. (2020); Yoon et al. (2022) under univariate time series setting. Many efforts have been made to defend against adversarial attack. Data augmentation has been widely applied in forecasting (Wen et al., 2020) and can improve model robustness. In the following section, we go beyond data augmentation and introduce more advanced techniques to enhance model robustness via randomized smoothing (Cohen et al., 2019) and mini-max defense using sparse layer.

4.1 Randomized Smoothing

Randomized smoothing is a post-training process and can be applied to any forecasting model $f_\theta(\mathbf{x})$, or $f(\mathbf{x})$ if the context is clear. Mathematically, let f be a random function that maps $\mathbf{x} \in \mathbb{R}^{d\times T}$ to a random vector $f(\mathbf{x})$ in $\mathbb{R}^d$ and denote the CDF of $f(\mathbf{x})$ as $F_{\mathbf{x}}(\mathbf{r}) = \mathbb{P}(f(\mathbf{x}) \preceq \mathbf{r})$, where $\preceq$ denotes element-wise inequality. Let $g_\sigma(\mathbf{x})$ be the randomized smoothing version of $f(\mathbf{x})$ with noise level σ and $g_\sigma(\mathbf{x})$ is also a random vector in $\mathbb{R}^d$ whose CDF is defined as

$$G_{\mathbf{x},\sigma}(\mathbf{r}) \;=\; \mathbb{E}_{\mathbf{z}\sim\mathcal{N}_{d\times T}(0,\sigma^2 I)}\left[\mathbb{P}\Big(f(\mathbf{x}+\mathbf{z}) \preceq \mathbf{r}\Big)\right] \;,$$

As will be shown later in Theorem 4.1, the smoothed forecaster $g_\sigma(\mathbf{x})$ has robustness certification. Different from Yoon et al. (2022), the $g_\sigma(\mathbf{x})$ here is a random vector rather than variable. Therefore, $G_{\mathbf{x},\sigma} : \mathbb{R}^d \to [0,1]$ is a multivariate CDF.

Robust Certificate. The next theorem certifies a Lipschitz continuity in terms of function L_∞-norm. Theorem 4.1 indicates that although the original CDF $F_{\mathbf{x}}(\mathbf{r})$ might not even be continuous, the smoothed CDF $G_{\sigma,\mathbf{x}}(\mathbf{r})$ is guaranteed to be Lipschitz continuous in $\mathbf{x}$, with Lipschitz constant scaling proportional to $\sqrt{d}$ and inverse-proportional to noise level σ.

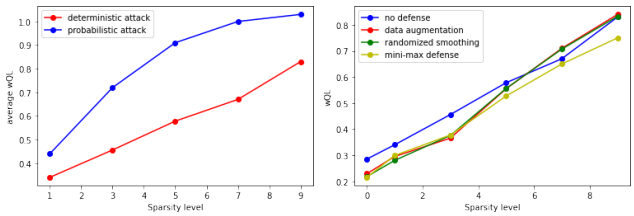
Theorem 4.1. *Let f be a random function that maps $\mathbf{x} \in \mathbb{R}^{d\times T}$ to a random vector $f(\mathbf{x})$ in $\mathbb{R}^d$ and denote the CDF of $f(\mathbf{x})$ as $F_{\mathbf{x}}(\mathbf{r}) = \mathbb{P}(f(\mathbf{x}) \preceq \mathbf{r})$. Let $g_\sigma(\mathbf{x})$ be the randomized smoothed version of $f(\mathbf{x})$, which is also a random vector in $\mathbb{R}^d$ whose CDF is defined as $G_{\mathbf{x},\sigma}(\mathbf{r}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\sigma I)}\mathbb{P}(f(\mathbf{x}+\mathbf{z}) \preceq \mathbf{r})$. Then for any $\mathbf{x} \in \mathbb{R}^{d\times T}$ and $\delta \in \mathbb{R}^{d\times T}$, we have*

$$\sup_{\mathbf{r}\in\mathbb{R}^d} |G_{\mathbf{x},\sigma}(\mathbf{r}) - G_{\mathbf{x}+\delta,\sigma}(\mathbf{r})| \le \frac{\sqrt{d}}{\sigma}\|\delta\|_F$$

Implementation. To get n future samples from randomized smoothing forecaster, we independently draw n isotropic Gaussian noises $\epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}_{d\times T}(0, \sigma^2 I)$ and compute the predicted distribution $f_\theta(\mathbf{x}(1+\epsilon_i))$ for future time series. For each $f_\theta(\mathbf{x}(1+\epsilon_i))$, draw a sample $\hat{\mathbf{z}}^{(i)} \sim f_\theta(\mathbf{x}(1+\epsilon_i))$ and collect $\hat{\mathbf{y}}^{(\mathbf{i})}$ for $i = 1, \ldots, n$. These will be the sample paths under randomized smoothing.

4.2 Mini-max Defense

We notice that our sparse layer can not only be used as an attacker, but is also helpful as a defense procedure.



(a) **(b)**

Figure 1: Plots of (a) averaged wQL under sparse indirect attack against the sparsity level on electricity dataset. The underlying model is a clean DeepVAR without defense. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$; and (b) average wQL under different defense mechanisms on electricity dataset. The attack type is deterministic attack. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$.

Formulation. We randomly initialize a sparse layer g_Θ with sparsity k as a hyper-parameter and a forecasting model f_θ from scratch. For each data point $\mathbf{x}$ in the training set, the sparse layer g_Θ is used to generate a sparse adversarial example $\hat{\mathbf{x}}$, which is then fed into f_θ to complete training phase. Specifically, in each epoch, the first step is to update the parameters of the sparse layer by maximizing the model's deviation from the true prediction $\ell_g = \sum_{i=1}^{n} \mathbb{E}_{\delta \sim g_\Theta(\mathbf{x}_i;k)} \mathbb{E}_{\mathbf{z}_i \sim f_\theta(\mathbf{x}_i+\delta)} \|\mathbf{z}_i - \mathbf{z}_i^{\text{true}}\|$, where y_i^{true} is the ground truth prediction. In the second step, we train the model f_θ on the corrupted examples generated by the current g_Θ. In other words, we update θ to maximize the model likelihood: $\ell_f = \sum_{i=1}^{n} \mathbb{E}_{\delta_i \sim g_\Theta(\mathbf{x}_i;k)} \log p_\theta(\mathbf{z}_i^{\text{true}} | \mathbf{x}_i + \delta_i)$. Note that g_Θ and f_θ compete over one another in the sense that in each epoch, g_Θ is trained to generate effective attack that could harm f_θ and f_θ is then trained to defend the attack from g_Θ. We call this defense mechanism a mini-max defense. Similar ideas have been exploited in deep generative models, such as GAN (Goodfellow et al., 2014a) and WGAN (Arjovsky et al., 2017). See Algorithm 2 for a detailed description.

Different from the sparse layer used in attack, this sparse layer in defense does not have access to the attack sparsity or the set of target time series I. Hence, we need to set the sparsity k as a hyper-parameter and skip the last step of the sparse layer described in Section 3.3 where we set $\delta_{I,*} = \mathbf{0}$.

5 Experiments

We conduct numerical experiments to demonstrate the effect of our proposed indirect sparse attack on a probabilistic DeepVAR model (Salinas et al., 2019) and compare various defense mechanisms including data augmentation, randomized smoothing and mini-max defense. The experiments are performed on standard real datasets for time series forecasting including Taxi (Taxi & Commission, 2015) and UCI Electricity (Asuncion & Newman, 2007) datasets preprocessed as in Salinas et al. (2020).

5.1 Experiment Setups

In empirical experiments, we target the prediction of the first time series at the last prediction time step, i.e. target time series $I = \{1\}$ and time horizon to attack $H = \{\tau\}$, the last time step, i.e., $\chi(\mathbf{z}) = x_{1,T+\tau}$. For the adversarial target $\mathbf{t}(\mathbf{x})$, we first draw a prediction $\hat{\mathbf{x}}$ from un-attacked model $p_\theta(\cdot|\mathbf{x})$ and choose $\mathbf{t} = c_1 \hat{x}_{1,T+\tau}$ for some $c_1 > 0$. Note that c_1 should be away from 1 to reflect adversarial target. The attack energy $\eta = c_2 \max|\mathbf{x}|$, is proportional to the largest element of the past observation in magnitude. Unless otherwise stated, the number of sample paths drawn from the prediction distribution $n = 100$ to quantify quantiles $q_{i,t}^{(\alpha)}$.

Dataset. Datasets for experiments include Electricity (Asuncion & Newman, 2007), Taxi (Taxi & Commission, 2015), Traffic (Asuncion & Newman, 2007), Solar (Lai et al., 2018), Wiki (Gasthaus et al., 2019). Check for more details on the datasets.

Forecaster For Electricity and Taxi datasets, we train a DeepVAR model implemented by pytorch-ts (Rasul, 2021) with target dimension 10 and rank 5. We choose $\tau = 24$ and $T = 4\tau = 96$, sparsity $k = 1, 3, 5, 7, 9$. In $\mathbf{t} = c_1 \hat{x}_{1,T+\tau}$ and $\eta = c_2 \max|\mathbf{x}|$, we select $c_1 = 0.5, 2.0$ and $c_2 = 0.5$ respectively and report the largest error produced by these choices of constants. We set attacking configuration $I = \{1\}$ and $H = \{\tau\}$.

Data augmentation and randomized smoothing Following the convention in Dang-Nhu et al. (2020); Yoon et al. (2022), we use relative noises in both data augmentation and randomized smoothing. That is, given a sequence of observation $\mathbf{x} = (x_{i,t})_{i,t} \in \mathbb{R}^{d\times T}$, we draw i.i.d. noise samples $\xi_{i,t} \sim \mathcal{N}(0, \sigma^2)$ and produce noisy input as $\tilde{x}_{i,t} \leftarrow x_{i,t}(1 + \xi_{i,t})$. In data augmentation, we train model with noisy input $\tilde{x}_{i,t}$. In randomized smoothing, the base model is still trained on noisy input $\tilde{x}_{i,t}$ with noise level σ. The noise level σ in the inference phase of randomized smoothing is chosen to be the same as that in data augmentation so there is no need to distinguish the σ used in the two processes. In all experiments, σ is set to 0.1.

Metrics We adopt weighted quantile loss (wQL) to measure the performance. (See Appendix E.)

5.2 Experiment Results

Electricity. The metrics under deterministic attack given by Algorithm 1 and probabilistic attack using sparse layer are reported in Table 1 and Table 2 respectively. Besides, we plot wQL under both attacks against sparsity level to better visualize the effect of different types of attack. See Figure 1a and Figure 1b.

Taxi. We report the performance of deterministic attack and probabilistic attack in Table 6 and Table 7 respectively in Appendix G. On taxi dataset, it can be observed that in most of the cases under both attacks, our mini-max defense mechanism achieves the best averaged wQL loss.

Message 1: Sparse and indirect attack is effective (as k increases) In the experiment, we can verify the effectiveness of sparse indirect attack, that is, one can attack the prediction of one time series without directly attacking the history of this time series. For example in Table 1, under deterministic attack, the average wQL is increased by 20% by only attacking one out of nine remaining time series (totally ten but the target time series is excluded). Moreover, attacking half of the time series can increase average wQL by 102%! This observation is even more noticeable under probabilistic attack: average wQL can be increased by 215% with 50% of the time series attacked. Besides, wQL loss increases as attack sparsity k increases, which is also an evidence that sparse indirect attack is effective.

Message 2: Prob. attack is more effective than det. one In general, average wQL increases as sparsity level increases and probabilistic attack appears to be more effective than deterministic one, see Figure 1a and Table 1. For example, under no defense when $k = 7$, probabilistic attack causes 50% larger wQL loss than deterministic one.

Message 3: RS and Minmax are more robust than data augmentation As can be seen in Figure 1b, Table 1 and Table 2, all three defense methods can bring robustness to the forecasting model. Data augmentation and randomized smoothing works well under small sparsity and mini-max defense achieves comparable performance as data augmentation and randomized smoothing under small sparsity and outperforms them under large sparsity.

Table 1: Average wQL on electricity dataset under **deterministic** attack. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

Sparsity	no defense	data augmentation	randomized smoothing	mini-max defense
no attack	0.2853±0.0825	0.2288±0.0792	0.2176±0.0700	**0.2154**±0.0705
1	0.3410±0.0946	0.2949±0.0716	**0.2826**±0.0718	0.2990±0.0772
3	0.4559±0.1344	**0.3655**±0.1097	0.3757±0.1012	0.3775±0.0923
5	0.5770±0.1772	0.5554±0.1636	0.5560±0.1751	**0.5273**±0.1558
7	0.6687±0.2131	0.7076±0.2321	0.7072±0.2308	**0.6506**±0.2111
9	0.8282±0.2847	0.8412±0.2896	0.8327±0.2786	**0.7503**±0.2588

5.3 Non-transferrablity of Attacks between Univaraite and Multivariate Forecasters

From the above Section 5.2, we verify the effectiveness of sparse indirect attack of multivariate forecasting models. In this subsection, we investigate the transferrability from univariate attack to

Table 2: Metrics on electricity dataset under **probabilistic** attack using sparse layer. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

Sparsity	no defense	data augmentation	randomized smoothing	mini-max defense
no attack	0.2909±0.0748	0.2374±0.0764	**0.2237**±0.0750	0.2342±0.0710
1	**0.4364**±0.1296	0.5923±0.0913	0.5940±0.1142	0.4935±0.1450
3	0.7245±0.2434	0.5738±0.1759	**0.4581**±0.1301	0.8079±0.2838
5	0.9143±0.3235	0.8422±0.2945	0.9276±0.3208	**0.5265**±0.1611
7	0.9991±0.3505	0.8267±0.2823	1.0100±0.3554	**0.6161**±0.1986
9	1.0317±0.3707	0.8139±0.2827	0.8919±0.3072	**0.6466**±0.2054

multivariate attack. To be specific, we study the question that if an attack is generated on the same subset (excluding target time series) of time series using a univariate model and then fed into a multivariate model, can it indirectly harm the prediction of target time series. We choose sparsity level $k = 1$ and other parameters are the same as what is described in Section 5.1. Algorithm 1 shows that the prediction of time series 1 is most sensitive to the history of time series 5. Thus, we use the technique in Dang-Nhu et al. (2020); Yoon et al. (2022) to generate univariate attack on time series 5 from DeepAR. Note that only the history of time series 5 has been adversely altered. The attacked time series is further fed into the same DeepVAR model.

Experiment result. The averaged wQL loss is reported in Table 3. For a better visualization, the history of time series 5 and prediction of time series 1 are plotted in Figure 4a and Figure 4b respectively. From the experiment results, attack transferred from univariate model doesn't serve as an effective indirect attack on multivariate model, which is also a reason why multivariate attack worth investigation.

Table 3: Transfer the attack from DeepAR to DeepVAR. Target items $I = \{1\}$ and time horizon to attack $H = \{\tau\}$. Clean DeepAR and DeepVAR models are used. Averaged wQL is reported below

No attack	Univariate attack	Multivariate attack
0.288	0.322	0.390

5.4 Additional Experiments Study

More datasets. We conduct additional experiments on Solar, Traffic and Wiki datasets. See Appendix for more details.

Ablation study. We also study the effect of hyper-parameters in our experiments. In this section, we investigate the effect of H by setting $H = \{\tau/2\}$ and $H = \{2\tau/3\}$. We also set $I = \{5\}$ and $I = \{10\}$ for targeting at different time series. We also choose the noise level σ in data augmentation and randomized smoothing from $\{0.2, 0.3\}$. Experiments details can be found in Appendix.

6 Conclusion

In this work, we investigate the existence of sparse indirect attack for multivariate time series forecasting models. We propose both deterministic approach and a novel probabilistic approach to finding effective adversarial attack. Besides, we adopt the randomized smoothing technique from image classification and univariate time series to our framework and design another mini-max optimization to effectively defend the attack delivered by our attackers. To the best of our knowledge, this is the first work to study sparse indirect attack on multivariate time series and develop corresponding defense mechanisms, which could inspire a future research direction.

References

Torben G Andersen, Tim Bollerslev, Peter Christoffersen, and Francis X Diebold. Volatility forecasting, 2005.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12):1694–1705, 2017.

Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2009.

Robert G Brown. Exponential smoothing for predicting demand. In *Operations Research*, volume 5, pp. 145–145. INST OPERATIONS RESEARCH MANAGEMENT SCIENCES 901 ELKRIDGE LANDING RD, STE . . . , 1957.

Ping-yeh Chiang, Michael Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection with median smoothing. *Advances in Neural Information Processing Systems*, 33:1275–1286, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

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

Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4724–4732, 2019.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pp. 1115–1124. PMLR, 2018.

Raphaël Dang-Nhu, Gagandeep Singh, Pavol Bielik, and Martin Vechev. Adversarial attacks on probabilistic autoregressive forecasting models. In *International Conference on Machine Learning*, pp. 2356–2365. PMLR, 2020.

Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing kalman filters for multivariate time series analysis. *Advances in Neural Information Processing Systems*, 33:2995–3007, 2020.

Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1901–1910. PMLR, 2019.

Sarah Gelper, Roland Fried, and Christophe Croux. Robust forecasting with exponential and Holt–Winters smoothing. *Journal of Forecasting*, 29(3):285–300, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Samuel Harford, Fazle Karim, and Houshang Darabi. Adversarial attacks on multivariate time series. *arXiv preprint arXiv:2004.00410*, 2020.

MJC Hu and Halbert E Root. An adaptive data processing system for weather forecasting. *Journal of Applied Meteorology and Climatology*, 3(5):513–523, 1964.

Kelvin Kan, François-Xavier Aubet, Tim Januschowski, Youngsuk Park, Konstantinos Benidis, Lars Ruthotto, and Jan Gasthaus. Multivariate quantile function forecaster. In *International Conference on Artificial Intelligence and Statistics*, pp. 10603–10621. PMLR, 2022.

Jongho Kim, Youngsuk Park, John D Fox, Stephen P Boyd, and William Dally. Optimal operation of a plug-in hybrid vehicle with battery thermal and degradation model. In *2020 American Control Conference (ACC)*, pp. 3083–3090. IEEE, 2020.

Aounon Kumar and Tom Goldstein. Center smoothing: Certified robustness for networks with structured outputs. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Bryan Lim, Stefan Zohren, and Stephen Roberts. Recurrent neural filters: Learning independent bayesian filtering steps for time series prediction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Linbo Liu and Danna Zhang. High-dimensional simultaneous inference on non-gaussian var model via de-biased estimator. *arXiv preprint arXiv:2111.01382*, 2021a.

Linbo Liu and Danna Zhang. Robust estimation of high-dimensional vector autoregressive models. *arXiv preprint arXiv:2109.10354*, 2021b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Gautam Raj Mode and Khaza Anuarul Hoque. Adversarial examples in deep learning for multivariate time series regression. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–10. IEEE, 2020.

Manfred Mudelsee. Trend analysis of climate time series: A review of methods. *Earth-science reviews*, 190:310–322, 2019.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Youngsuk Park, Kanak Mahadik, Ryan A Rossi, Gang Wu, and Handong Zhao. Linear quadratic regulator for resource-efficient cloud services. In *Proceedings of the ACM Symposium on Cloud Computing*, pp. 488–489, 2019.

Youngsuk Park, Danielle Maddix, François-Xavier Aubet, Kelvin Kan, Jan Gasthaus, and Yuyang Wang. Learning quantile functions without quantile crossing for distribution-free time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 8127–8150. PMLR, 2022.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

Kashif Rasul. PytorchTS, 2021. URL https://github.com/zalandoresearch/pytorch-ts.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

NYC Taxi and Limousine Commission. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2015.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Di Wang and Ruey S Tsay. Robust estimation of high-dimensional vector autoregressive models. *arXiv preprint arXiv:2107.11002*, 2021.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pp. 6607–6617. PMLR, 2019.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Lily Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. Proven: Verifying robustness of neural networks with a probabilistic approach. In *International Conference on Machine Learning*, pp. 6727–6736. PMLR, 2019.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

TaeHo Yoon, Youngsuk Park, Ernest K Ryu, and Yuyang Wang. Robust probabilistic time series forecasting. *arXiv preprint arXiv:2202.11910*, 2022.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

Jindi Zhang, Yang Lou, Jianping Wang, Kui Wu, Kejie Lu, and Xiaohua Jia. Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles. *arXiv preprint arXiv:2108.02940*, 2021.

A ILLUSTRATION OF ATTACK ON FINANCIAL INSTITUTE

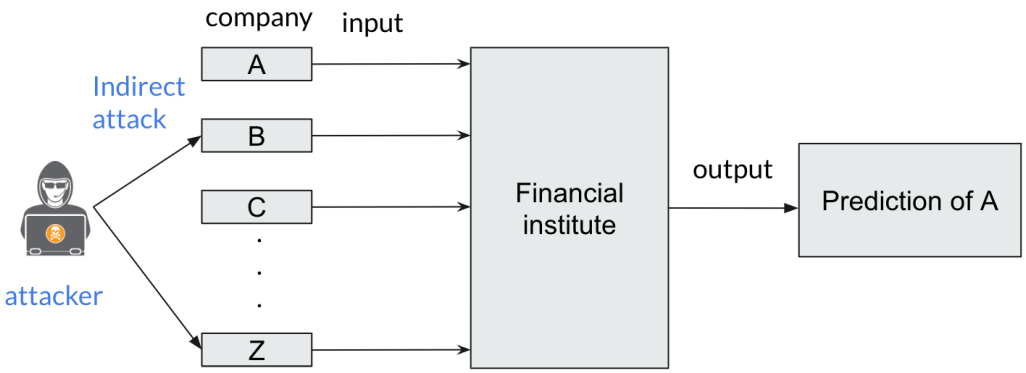
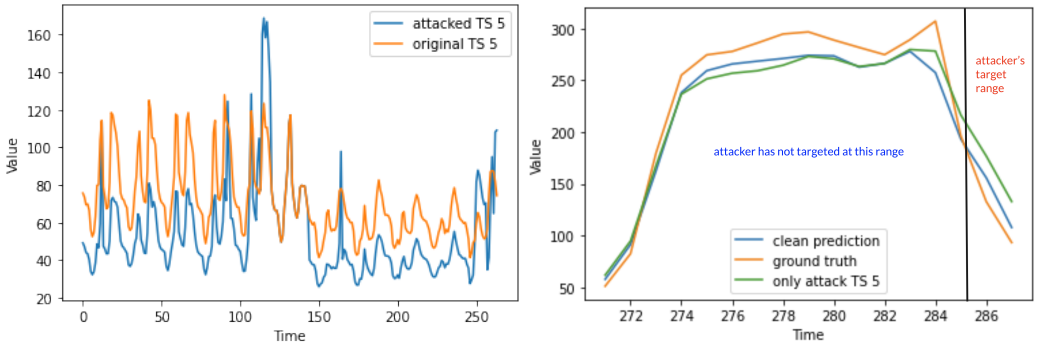


Figure 2: Illustration of indirect stealth attack to financial institute. The attacker targets at adversely altering the prediction of company A's stock price. To perform indirect attack, the attacker selected to attack company B and company Z.

B ADVERSARIAL ATTACK EXAMPLES



(a) **(b)**

Figure 3: Plots of (a) authentic (orange) and perturbed (blue) versions of time-series (TS) 5, which is selected by an attacker to mount an indirect attack on TS 1; and (b) ground-truth (orange), no-attack (blue) and under-attack (green) predictions for TS 1. No alteration was made to TS 1 but the value of TS 1 at the attack time step ($t = 288$) were adversely altered in the under-attack (green) setting, which can set the prediction of TS 1 significantly away from the ground truth.

C ALGORITHMS

C.1 DETERMINISTIC ATTACKING ALGORITHM

C.2 MINI-MAX DEFENSE ALGORITHM

D DATASETS

- Electricity: consists of hourly electricity consumption time series from 370 customers.
- Taxi: traffic time series of New York taxi rides taken at 1214 locations for every 30 minutes from January 2015 to January 2016 and considered to be heterogeneous. We use the taxi-30min dataset provided by GluonTS.
- Traffic: hourly occupancy rate, between 0 and 1, of 963 San Francisco car lanes.

Algorithm 1 Deterministic sparse attack algorithm

Input: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{d\times T}$, $f_\theta(\mathbf{x})$, back-test target $\mathbf{z} = (\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+\tau})$. Future time horizon $H \subseteq [\tau]$. Target time series $I \subseteq [d]$. Attack budget η. Sparsity k. Number of iteration N.
Output: Sparse attack $\delta \in \mathbb{R}^{d\times T}$ with row sparsity k and $\delta_{I,*} = \mathbf{0}$.
1. Initialize $\delta = 0 \in \mathbb{R}^{d\times T}$
2. Draw a predicted sample $\hat{\mathbf{z}}$ from $p_\theta(\mathbf{z}|\mathbf{x})$. Get adversarial target value $\mathbf{t} = \chi(c\hat{\mathbf{z}})$.
for iteration $= 1, \ldots, N$ **do**
3. Compute predicted distribution for $\mathbf{z}$: $p(\mathbf{z}|\mathbf{x}+\delta)$.
4. Compute expected loss under targeted attack:

$$\ell = \sum_{h\in H, i\in I} \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x}+\delta)}(z_{i,T+h} - t_{i,T+h})^2.$$

5. Use any first order method to update δ so as to minimize ℓ.
6. Clip δ with threshold η: $\delta_{i,t} = \delta_{i,t}\min\{1, \eta/|\delta_{i,t}|\}$
end for
7. For each time series $i \notin I$, compute cumulative perturbation over all time: $c_i = \sum_{t=1}^T \delta_{it}$ and sort c_i in descending order: $c_{\pi(1)} \geq c_{\pi(2)} \geq \cdots \geq c_{\pi(d)}$.
8. Keep $\delta_{\pi(1),*}, \ldots, \delta_{\pi(k),*}$ and set $\delta_{\pi(k+1),*}, \ldots, \delta_{\pi(d),*} = 0$.
9. Output δ.

Algorithm 2 Mini-max defense algorithm

Input: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{d\times T}$. Forecasting dataset $\mathcal{D} = \{\mathbf{X}_i, \mathbf{z}_i^{\text{true}}\}_{i=1}^n$, where $\mathbf{X}_i \in \mathbb{R}^{d\times T}$ and $\mathbf{z}_i^{\text{true}} \in \mathbb{R}^{d\times\tau}$ is obtained from back-test window. Sparsity k in the sparse layer $g_\Theta(\mathbf{x}; k)$
Output: A forecasting model $f_\theta(\mathbf{x})$.
for epoch $= 1, \ldots, N$ **do**
3. Compute loss for the sparse layer $g_\Theta(\mathbf{x}; k)$:

$$\ell_g = -\sum_{i=1}^n \mathbb{E}_{\delta\sim g_\Theta(\mathbf{X}_i;k)}\mathbb{E}_{\mathbf{z}_i\sim f_\theta(\mathbf{X}_i+\delta)}\|\mathbf{z}_i - \mathbf{z}_i^{\text{true}}\|.$$

4. Update Θ in the sparse layer to minimize ℓ_g.
5. Let $p(\cdot|\mathbf{x})$ be the output distribution of $f_\theta(\mathbf{x})$. Compute likelihood for model $f_\theta(\mathbf{x})$:

$$\ell_f = \sum_{i=1}^n \mathbb{E}_{\delta_i\sim g_\Theta(\mathbf{X}_i;k)} \log p_\theta(\mathbf{z}_i^{\text{true}}|\mathbf{X}_i + \delta_i).$$

6. Update θ in forecasting model to maximize ℓ_f.
end for

- Solar: hourly photo-voltaic production of 137 stations in Alabama State used in Lai et al. (2018).
- Wiki: daily page views of 2000 Wikipedia pages used in Gasthaus et al. (2019).

E METRICS

We measure the performance of model under attacks by the popular metric especially for probabilistic forecasting models: weighted quantile loss (wQL), which is defined as

$$\text{wQL}(\alpha) = 2\frac{\sum_{i,t}[\alpha\max(x_{i,t} - q_{i,t}^{(\alpha)}, 0) + (1-\alpha)\max(q_{i,t}^{(\alpha)} - x_{i,t}, 0)]}{\sum_{i,t}|x_{i,t}|},$$

where $\alpha \in (0, 1)$ is a quantile level. In practical application, under-prediction and over-prediction may cost differently, suggesting wQL should be one's main consideration especially for probabilistic

forecasting models. In the subsequent sections, we calculate average wQL over a range of $\alpha = [0.1, 0.2, \ldots, 0.9]$ and evaluate the performance in terms of averaged wQL.

F MORE METRICS ON ELECTRICITY DATASET

To measure the performance of a forecasting model, other metrics like Weighted Absolute Percentage Error (WAPE) or Weighted Squared Error (WSE) are also considered by a large body of literature. For completeness, we present the definition of WAPE and WSE:

$$\text{WAPE} = \sum \left| \frac{\text{predicted value}}{\text{true value}} - 1 \right| = \frac{1}{|I||H|} \sum_{i \in I, h \in H} \left| \frac{\frac{1}{n}\sum_{j=1}^{n} \hat{x}^j_{T+h,i}}{x_{T+h,i}} - 1 \right|$$

$$\text{WSE} = \sum \left(\frac{\text{predicted value}}{\text{true value}} - 1 \right)^2 = \frac{1}{|I||H|} \sum_{i \in I, h \in H} \left(\frac{\frac{1}{n}\sum_{j=1}^{n} \hat{x}^j_{T+h,i}}{x_{T+h,i}} - 1 \right)^2$$

We report WAPE, WSE and wQL under deterministic and probabilistic attacks on electricity dataset in Table 4 and Table 5.

Table 4: Metrics on electricity dataset under **deterministic** attack. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

	no defense			data augmentation			randomized smoothing			mini-max defense		
Sparsity	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE	WSE	wQL
no attack	0.4005±0.2036	0.2360±0.2525	0.2991±0.1684	0.4241±0.2092	0.2596±0.2625	0.3280±0.1497	0.3501±0.1630	0.1710±0.1486	0.2751±0.1068	0.3237±0.1379	0.1394±0.0913	0.2342±0.0917
1	0.4900±0.2488	0.3529±0.3769	0.3745±0.2106	0.4123±0.1829	0.2310±0.1934	0.3019±0.1138	0.4209±0.1700	0.2298±0.1683	0.2965±0.1003	0.4498±0.2253	0.2949±0.2276	0.3511±0.1825
3	0.6382±0.3434	0.6222±0.5886	0.5043±0.2917	0.5654±0.2475	0.4313±0.3707	0.3919±0.1876	0.5887±0.2543	0.4644±0.3784	0.3965±0.1797	0.7447±0.3758	0.8120±0.6684	0.6038±0.3358
5	0.7524±0.3675	0.8123±0.6218	0.6097±0.3218	0.7460±0.3803	0.8201±0.6628	0.5379±0.2833	0.7504±0.3607	0.8002±0.5999	0.5619±0.2779	0.9603±0.4190	1.2419±0.8369	0.8182±0.3845
7	0.8786±0.4171	1.0889±0.7785	0.7432±0.3702	0.8465±0.4014	1.0102±0.6389	0.6425±0.2985	0.8353±0.4315	1.0369±0.7496	0.6311±0.3152	1.1056±0.4847	1.6504±1.0591	0.9689±0.4350
9	1.0134±0.4541	1.4028±0.9685	0.8851±0.4023	0.9093±0.4454	1.1883±0.7720	0.7007±0.3395	0.9986±0.5026	1.4574±0.9998	0.7700±0.3717	1.2476±0.4860	1.9870±1.0815	1.1133±0.4306
full attack	1.2449±0.5522	2.1055±1.4686	1.1031±0.5002	1.0609±0.4661	1.5214±0.8815	0.8188±0.3650	1.1221±0.5100	1.7331±0.9988	0.8959±0.3988	1.2587±0.4989	2.0380±1.1256	1.1246±0.4471

Table 5: Metrics on electricity dataset under **probabilistic** attack using sparse layer. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

	No defense			data augmentation			randomized smoothing			
Sparsity	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE
no attack	0.3842±0.2620	0.2162±0.3044	0.2909±0.0748	0.3074±0.1746	0.1250±0.0946	0.2374±0.0764	0.2858±0.1547	0.1056±0.0761	**0.2237**±0.0750	0.3218±0.142
1	0.6230±0.6324	0.7881±1.1864	**0.4364**±0.1296	0.7476±0.7240	1.0830±1.8593	0.5923±0.0913	0.7683±0.8771	1.3596±2.7290	0.5940±0.1142	0.6990±0.695
3	1.0540±0.7522	1.6768±1.4810	0.7245±0.2434	0.8484±0.6809	1.1834±1.3998	0.5738±0.1759	0.6784±0.5230	0.7337±0.7698	**0.4581**±0.1301	0.9909±0.756
5	1.2078±0.7451	2.0139±2.0667	0.9143±0.3235	1.1444±0.6665	1.7538±1.4318	0.8422±0.2945	1.2310±0.7025	2.0090±1.6609	0.9276±0.3208	0.6966±0.455
7	1.3236±0.7310	2.2863±1.8336	0.9991±0.3505	1.1304±0.6522	1.7031±1.4053	0.8267±0.2823	1.3496±0.6777	2.2809±1.7240	1.0100±0.3554	0.8424±0.780
9	1.3656±0.8671	2.6166±2.6679	1.0317±0.3707	1.0912±0.6181	1.5727±1.2081	0.8139±0.2827	1.1978±0.6742	1.8894±1.5309	0.8919±0.3072	0.8691±0.741

G ADDITIONAL EXPERIMENTS ON TAXI DATASET

In this section, we report experiment results on Taxi dataset.

Table 6: Metrics on taxi dataset under **deterministic** attack. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

	no defense			data augmentation			randomized smoothing			
Sparsity	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE	WSE	wQL	W
no attack	3.1753±0.6548	16.3328±6.8222	1.2135±0.4050	3.4020±0.6503	17.7390±7.2134	1.2137±0.4091	3.4214±0.6501	17.8661±7.0561	1.2574±0.4281	2.936
1	3.2884±0.6591	17.1469±6.6485	1.3152±0.4580	3.5884±0.6577	19.1832±7.1234	1.3455±0.4666	3.6060±0.6559	19.2739±7.1416	1.3455±0.4627	2.977
3	3.8517±0.7110	22.2038±7.4880	1.6389±0.5810	4.1630±0.7086	24.6490±8.2418	1.6805±0.5982	4.1148±0.6872	23.8149±7.6266	1.6503±0.5756	3.344
5	4.5853±0.8062	30.5002±9.4037	2.0317±0.7161	4.8419±0.7837	32.3970±9.5921	2.0625±0.7290	4.7912±0.7643	31.4706±9.2416	2.0123±0.7059	3.948
7	5.2952±0.8884	39.5429±10.9774	2.3695±0.8064	5.5116±0.9026	42.2533±11.8199	2.3712±0.8028	5.3876±0.8831	40.3946±11.2429	2.3450±0.7978	4.516
9	5.7671±0.9517	46.4608±12.3892	2.5605±0.8531	5.8631±0.9769	48.2877±13.5654	2.5525±0.8616	5.8490±0.9610	47.6729±13.2402	2.5422±0.8619	5.056
full attack	5.7407±0.9536	46.2118±12.1696	2.6222±0.8798	5.7618±0.9307	45.8250±12.4133	2.5579±0.8795	5.6389±0.9092	43.8469±11.5653	2.5403±0.8761	4.462

Table 7: Metrics on taxi dataset under **probabilistic** attack. Target time series $I = \{1\}$ and attacked time stamp $H = \{\tau\}$. Smaller is better.

	no defense			data augmentation			randomized smoothing			
Sparsity	WAPE	WSE	wQL	WAPE	WSE	wQL	WAPE	WSE	wQL	WAP
no attack	3.2019±0.6620	16.6407±7.0108	1.2118±0.4412	3.4348±0.6568	18.0856±7.3327	1.2526±0.4733	3.3940±0.6477	17.6348±7.0878	1.2241±0.4531	2.9964±0
1	3.5098±0.7117	19.7020±7.1520	1.4598±0.5315	3.5353±0.6450	18.5629±6.7227	1.3539±0.5199	3.5961±0.6397	18.8968±6.6969	1.3512±0.5100	3.0503±0
3	3.5043±0.7429	20.3267±8.1305	1.5659±0.6589	3.7547±0.7278	21.8185±7.6430	1.5446±0.6197	3.8490±0.7015	21.9873±7.0132	1.5567±0.5784	2.9650±0
5	4.2285±0.7365	25.7872±7.6545	1.9123±0.7513	3.9991±0.7031	23.1989±6.9567	1.7824±0.6962	4.1910±0.7303	25.3399±7.4696	1.8857±0.7441	2.9543±0
7	4.7813±0.8095	32.4134±9.1638	2.2915±0.8954	3.8031±0.7747	23.2123±8.9699	1.7340±0.7638	4.3014±0.7394	26.4713±9.5438	1.8370±0.7597	3.1284±0
9	5.3666±0.8732	39.9142±10.3507	2.4815±0.9286	5.0260±0.7627	33.7399±8.1455	2.1159±0.7515	5.3652±0.8100	38.3500±9.4049	2.2400±0.7860	3.1476±0

H DETAILED PROOFS

Proof of Theorem 3.1. We can compute

$$\begin{aligned}\mathbb{P}(\delta_{i,*} = \mathbf{0}) &= 1 - \mathbb{P}\Big(u_i \le \Phi^{-1}\Big(r_i(\gamma)\Big)\Big) \\ &= 1 - r_i(\gamma) \end{aligned} \tag{H.1}$$

That is, with probability $1 - r_i(\gamma)$, $\delta_{i,*} = 0$. Equivalently, $\delta_{i,*}$ is distributed by a degenerated probability measure with Dirac density $D(\delta_{i,*})$ concentrated at 0. On the other hand, with probability $r_i(\gamma)$, $\delta_{i,*}$ is distributed as $q_i'(\cdot|\mathbf{x};\beta)$. Combining the two cases, it follows that $\delta_{i,*}$ is distributed by a mixture of $q_i'(\cdot|\mathbf{x};\beta)$ and $D(\delta_{i,*})$ with weights $r_i(\gamma)$ and $1 - r_i(\gamma)$ respectively. □

Proof of Theorem 3.2. By the construction of $r_i(\gamma)$,

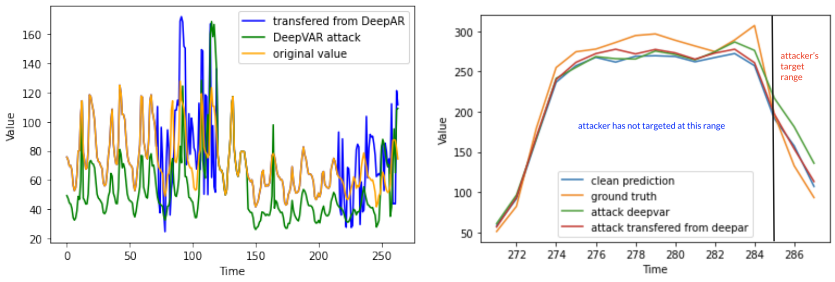
$$\begin{aligned}\mathbb{E}\Big[s(\delta)\Big] &= \sum_{i=1}^{d} \mathbb{E}\left[\mathbb{I}\left(u_i \le \Phi^{-1}\left(r_i(\gamma)\right)\right)\right] \\ &= \sum_{i=1}^{d} \mathbb{P}\left(u_i \le \Phi^{-1}\left(r_i(\gamma)\right)\right) \\ &= \sum_{i=1}^{d} r_i(\gamma) = \frac{k}{\sqrt{d}} \cdot \frac{\sum_{i=1}^{d} \gamma_i^{1/2}}{\left(\sum_{i=1}^{d} \gamma_i\right)^{1/2}} \le k\end{aligned}$$

□

Proof of Theorem 4.1. Denote $p_\sigma(\cdot)$ as the density of $\mathcal{N}(0, \sigma I_d)$ and $p(\cdot)$ as the density of $\mathcal{N}(0, I_d)$. Consider

$$\begin{aligned}\sup_{\mathbf{r}\in\mathbb{R}^d} |G_{\mathbf{x},\sigma}(\mathbf{r}) - G_{\mathbf{x}+\delta,\sigma}(\mathbf{r})| &= \sup_{\mathbf{r}\in\mathbb{R}^d} \left| \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} \big(F_{\mathbf{x}+\mathbf{z}}(\mathbf{r}) - F_{\mathbf{x}+\delta+\mathbf{z}}(\mathbf{r})\big) p_\sigma(\mathbf{z})\,\mathrm{d}\mathbf{z} \right| \\ &= \sup_{\mathbf{r}\in\mathbb{R}^d} \left| \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} F_{\mathbf{z}}(\mathbf{r}) \big(p_\sigma(\mathbf{z}-\mathbf{x}) - p_\sigma(\mathbf{z}-\mathbf{x}-\delta)\big)\,\mathrm{d}\mathbf{z} \right| \\ &= \sup_{\mathbf{r}\in\mathbb{R}^d} \left| \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} \int_0^1 F_{\mathbf{z}}(\mathbf{r}) \nabla p_\sigma(\mathbf{z}-\mathbf{x}-t\delta)\delta\,\mathrm{d}t\,\mathrm{d}\mathbf{z} \right| \\ &= \sup_{\mathbf{r}\in\mathbb{R}^d} \left| \int_0^1 \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} F_{\mathbf{z}}(\mathbf{r}) \left(\delta \cdot \frac{\mathbf{z}-\mathbf{x}-t\delta}{\sigma^2}\right) p_\sigma(\mathbf{z}-\mathbf{x}-t\delta)\,\mathrm{d}\mathbf{z}\,\mathrm{d}t \right| \\ &= \frac{1}{\sigma} \sup_{\mathbf{r}\in\mathbb{R}^d} \left| \int_0^1 \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} F_{\mathbf{x}+t\delta+\mathbf{z}}(\mathbf{r}) \left(\delta \cdot \mathbf{z}\right) p(\mathbf{z})\,\mathrm{d}\mathbf{z}\,\mathrm{d}t \right| \\ &\le \frac{1}{\sigma} \int_{\mathbf{z}\in\mathbb{R}^{d\times T}} |\delta \cdot \mathbf{z}| p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\le \frac{\|\delta\|_2}{\sigma} \left(\mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,I_d)} \|\mathbf{z}\|_2^2\right)^{1/2} = \frac{\sqrt{d}}{\sigma} \|\delta\|_2,\end{aligned}$$

which completes the proof. □



(a) **(b)**

Figure 4: Plots of (a) authentic (orange), DeepAR-attacked (blue) and DeepVAR-attacked (green) versions of time-series (TS) 5; and (b) ground-truth (orange), no-attack (blue), under-DeepAR-attack (red) and under-DeepVAR-attack (green) predictions for TS 1. Compared to clean prediction, the value of TS 1 at the attack time step ($t = 288$) were adversely altered by DeepVAR-attack (green) but only slightly altered by DeepAR-attack (red).

I NON-TRANSFERRABILITY OF ATTACKS BETWEEN UNIVARIATE AND MULTIVARIATE FORECASTERS

In this section, we present the figure that illustrates the experiment results of univariate attack and multivaraite attack.