Online Evasion Attacks on Recurrent Models: The Power of Hallucinating the Future

Byunggill Joe[1] , **Insik Shin**[1] and **Jihun Hamm**[2*]

[1] School of Computing, KAIST, Daejeon, South Korea
[2] Department of Computer Science, Tulane University, Louisiana, USA
{byunggill.joe, insik.shin}@kaist.ac.kr, jhamm3@tulane.edu

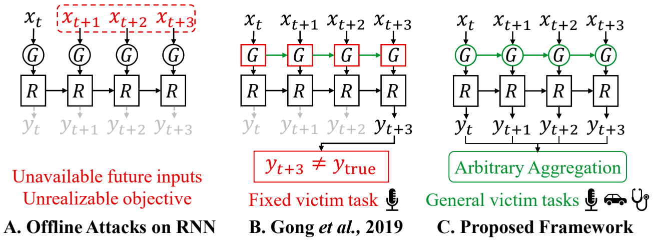
Abstract

Recurrent models are frequently being used in online tasks such as autonomous driving, and a comprehensive study of their vulnerability is called for. Existing research is limited in generality only addressing application-specific vulnerability or making implausible assumptions such as the knowledge of future input. In this paper, we present a general attack framework for online tasks incorporating the unique constraints of the online setting different from offline tasks. Our framework is versatile in that it covers time-varying adversarial objectives and various optimization constraints, allowing for a comprehensive study of robustness. Using the framework, we also present a novel white-box attack called Predictive Attack that 'hallucinates' the future. The attack achieves 98 percent of the performance of the ideal but infeasible clairvoyant attack on average. We validate the effectiveness of the proposed framework and attacks through various experiments.

1 Introduction

Deep neural networks (DNN) are discovered to be surprisingly vulnerable to imperceptibly small input noises [Szegedy *et al.*, 2014; Goodfellow *et al.*, 2015]. Many different types of vulnerabilities of DNNs have been demonstrated in various tasks, including classification, regression, and generative methods [Mądry *et al.*, 2018; Gondim-ribeiro *et al.*, 2018; Dang-Nhu *et al.*, 2020]. Most attacks have focused on offline evasion attacks on non-temporal models, such as the image classification model, where an attacker has access to the whole input example, such as an image.

However, there are many security-critical tasks that involve recurrent neural networks (RNN) performed in an online fashion [Suradhaniwar *et al.*, 2021; Pinto *et al.*, 2021; Huang *et al.*, 2020]. For instance, mortality prediction [Harutyunyan *et al.*, 2019] continuously monitors hospitalized patients for early warning, and an autonomous driving agent that uses sensors to decide the steering angle of the vehicle [Kiran *et al.*, 2021]. Unlike the offline setting, an attacker cannot observe the entire input sequence because inputs arrive as a real-time stream to a victim and an attacker.



A. Offline Attacks on RNN **B. Gong *et al.*, 2019** **C. Proposed Framework**

Figure 1: Framework comparison for attacking an RNN of an online task. "G": Attack perturbation generator, "R": Victim RNN.

Previous works [Xie *et al.*, 2020; Fawaz *et al.*, 2019; Dang-Nhu *et al.*, 2020; Oregi *et al.*, 2018] evaluated vulnerabilities of RNNs but they implicitly assumed that future inputs are observable, which is implausible (Figure 1-A, dashed box). Meanwhile, [Gong *et al.*, 2019] introduced a framework for real-time attack (Figure 1-B) with two constraints unique to the online problem: 1) future inputs are unobservable, and 2) the past inputs are unchangeable. However, the framework is rather specific to speech recognition (Figure 1-B, red boxes), where classification is performed only once after receiving the entire speech. Such an approach is inapplicable to dynamic online tasks such as autonomous driving, where the agent has to continuously decide the steering angle.

In this paper, we propose a more general framework of online evasion attacks[1] allowing a victim recurrent model to make continuous predictions or decisions (Figure 1-C, circled G and a round box). Our framework can accommodate various adversarial objectives on RNN to address different attack scenarios. In particular, our framework makes time-varying adversarial objectives possible, unlike previous approaches to attack at a specific time. The versatility of our framework will allow for a comprehensive robustness study of recurrent models. To showcase of the versatility, we reformulate the objective of real-time attack [Gong *et al.*, 2019], and present novel adversarial objectives such as Time-window and Surprise objectives using our framework.

As an effective solution to our framework, we propose a

[*] Corresponding author.

[1] https://github.com/byunggilljoe/rnn_online_evasion_attack. Appendix is included.

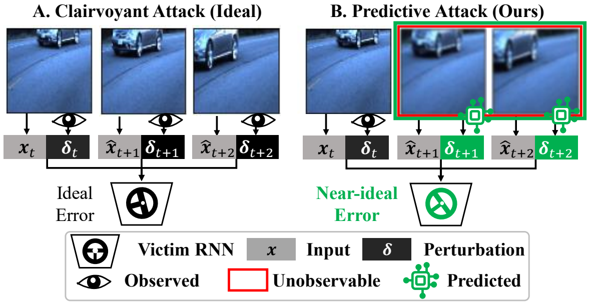


Figure 2: Attacking a vision-based autonomous driving agent to change the steering angles. Clairvoyant Attack (A) can see all future inputs and achieves the best attack but it is unrealizable. Predictive Attack (B) emulates the clairvoyant by hallucinating the future using a predictive model of the input sequence.

novel white-box attack called Predictive Attack (Figure 2-B). An ideal solution to the online problem is the clairvoyant attack (Figure 2-A), where an attacker does not suffer from the online constraints, foreseeing the entire future input. Thus the clairvoyant attack can find attack perturbations with existing offline methods [Mądry *et al.*, 2018; Croce and Hein, 2020]. Instead of clairvoyance, Predictive Attack 'hallucinate' the future (Figure 2-B, green box) with a trained predictive model of input sequences, mimicking the crystal ball of a clairvoyant. Since accurate prediction can be difficult, we propose an additional alternative attack called IID Attack that replaces accurate prediction with IID sampling. They perform surprisingly well, and we ascribe this to the importance of considering the hidden states and input orders when attacking recurrent models.

We evaluate our attacks using six datasets. Our predictive attack approaches 98% of the performance of the clairvoyant on average. We perform further empirical analysis of the predictive attacks demonstrating the versatility of our framework and attack robustness. We summarize our contributions as follows.

- We introduce a general formulation of online evasion attacks on recurrent models, which can accommodate various types of attack objectives and constraints, allowing for a comprehensive study of robustness.
- We propose two novel white-box attacks, Predictive Attack and IID Attack, based on hallucination of the future to emulate the ideal clairvoyant attacker.
- We evaluate the performance of our attacks under various conditions using real-world data and demonstrate the versatility and robustness of our framework and attacks.

2 Setting

Inputs and Outputs. An input stream/sequence $\boldsymbol{x}$ of length L is a sequence of n-dimensional vectors $(x_1, x_2, ..., x_L) \in \mathbb{R}^{L\times n}$ where the index refers to time step. Similarly, the output sequence $\boldsymbol{y}$ of length L is sequence of outputs $(y_1, y_2, ..., y_L)$ where $y_i \in \mathbb{R}$ for a regression problem and $y_i \in \{1, ..., C\}$ for a classification problem.

Victim Task. We attack recurrent neural networks (RNN) that continuously predict the output at each time step. Formally, an RNN is a pair of functions f_θ and g_θ. At time t, $f_\theta : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ predicts the current output by $y_t = f_\theta(x_t, h_t)$ using the current input x_t and the hidden state $h_t \in \mathbb{R}^m$. The dynamics of the RNN is determined by $g_\theta : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ which maps (x_t, h_t) to the next hidden state by $h_{t+1} = g_\theta(x_t, h_t)$.

Threat Model. We assume attackers have white-box access to a victim model. Attackers can define an attack objective and loss and can compute the derivative of the loss with respect to an input. Also, Attackers have access to some examples of input streams.

3 Online Evasion Attack Framework

Problem. The attacker aims to mislead a victim RNN model (f_θ, g_θ) to output the (adversarial) target labels or values $(y_1^a, \cdots, y_L^a)$ by using the perturbed input sequence $(x_1 + \delta_1, \cdots, x_L + \delta_L)$. This is done by minimizing[2] the aggregate value of the losses $\mathcal{L}_1^{\text{adv}}, \cdots, \mathcal{L}_L^{\text{adv}}$:

$$\boldsymbol{\delta} = \underset{\boldsymbol{\delta}=(\delta_1,\cdots,\delta_L)\in\Delta}{\arg\min} \ \text{Agg}\left(\mathcal{L}_1^{\text{adv}}, \cdots, \mathcal{L}_L^{\text{adv}}\right), \text{ where} \quad (1)$$

$\mathcal{L}_i^{\text{adv}}$ is the loss at time i: $\mathcal{L}_i^{\text{adv}} = \mathcal{L}(f_\theta(x_i + \delta_i, h_i^\delta), y_i^a)$, and h_i^δ is the hidden state of the RNN at time i:

$$h_i^\delta = g_\theta(x_{i-1} + \delta_{i-1}, h_{i-1}^\delta). \quad (2)$$

The $\text{Agg}(\cdot)$ refers to a method of temporal aggregation, and Δ refers to any constraint on the perturbation sequence. Compared to previous work [Gong *et al.*, 2019], the present formulation (Equation 1∼2) is much more flexible since the loss and the target are allowed to be time-varying. For concreteness, we will use the temporal summation $\text{Agg}(\mathcal{L}_1, \cdots, \mathcal{L}_l) = \sum_{i=1}^{L} \mathcal{L}_i$ and the ℓ_p constraint $\Delta = \{\|\delta_i\|_p \le \epsilon, \quad \forall i\}$ by default:

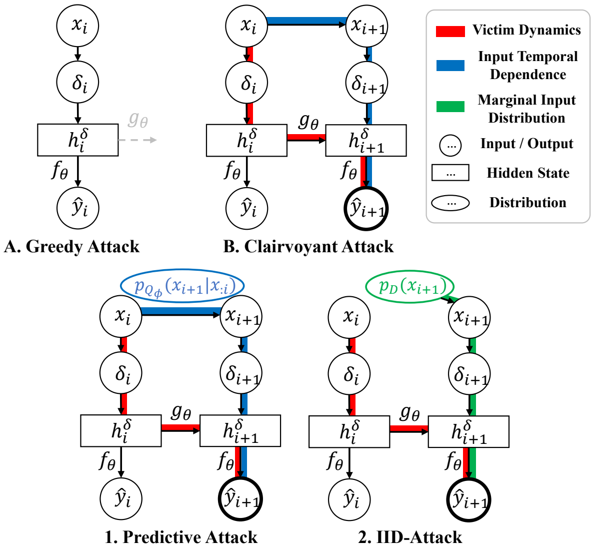
$$\boldsymbol{\delta} = \underset{\|\delta_i\|_p\le\epsilon,\ \forall i}{\arg\min} \sum_{i=1}^{L} \mathcal{L}^{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta). \quad (3)$$

Online Constraints. Critically different from the much-studied offline attacks, an online attack has to follow physical constraints [Gong *et al.*, 2019]. Firstly, an attacker cannot perturb the future or the past input but only the current input. Therefore, to solve Equation 3 an attacker has to solve

$$\delta_t = \underset{\|\delta_t\|_p\le\epsilon}{\arg\min} \sum_{i=t}^{L} \mathcal{L}^{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta) \quad (4)$$

at each time step $t = 1, \cdots, L$, which is the core of the general online attack. Since the losses of the past $(i < t)$ are unchangeable they do not appear in the sum of the losses. An important thing to note is that the current perturbation δ_t affects all future losses $\mathcal{L}_{t+1}^{\text{adv}}, \mathcal{L}_{t+2}^{\text{adv}}, \cdots$ due to the nature of RNNs, which we call **victim model dynamics** property. A

[2] The current description is for targeted attacks but we can also perform untargeted attacks as well.



A. Greedy Attack

B. Clairvoyant Attack

1. Predictive Attack

2. IID-Attack

Figure 3: Comparison of attack methods: reference (A and B) and proposed (1, 2). Please refer to Method for details.

successful online attack therefore has to exploit this property. Furthermore, the sum can be rewritten as

$$\delta_t = \arg\min_{\|\delta_t\|_p \le \epsilon} \underbrace{\mathcal{L}^{\text{adv}}(x_t, y_t^a, \delta_t, h_t^\delta)}_{\text{(A) Current}} + \sum_{i=t+1}^{L} \underbrace{\mathcal{L}^{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta)}_{\text{(B) Future, not observed.}} . \tag{5}$$

As the equation shows, this optimization problem cannot be solved directly due to the second constraint of the online attack: we do not know the future inputs $x_{t+1}, x_{t+2}, \cdots$. Although seemingly impossible, we make it possible by exploiting the **temporal dependence** of the inputs in a stream, on which the decisions of RNN depend ultimately. (Details in the next section.) To mount a successful attack, an online attacker has to exploit both victim model dynamics and temporal dependence.

3.1 Greedy and Clairvoyant Attack

We propose two reference attacks that exemplify a crude attack and an ideal attack. These attacks can help us understand the other attacks in the following sections. **Greedy Attack** (Figure 3-A) provides a lower bound of attack performance which does not consider the victim model dynamics and the temporal dependency. This attack only considers the current loss (A) of Equation 5. **Clairvoyant Attack** (Figure 3-B) is an ideal, unrealizable attack that assumes the full observability of the future part of an input sequence; thus can fully use the victim model dynamics (Figure 3-B, red line), and temporal dependence (Figure 3-B, blue line). The Clairvoyant provides the upper bound of performance an attack can achieve.

4 Method

4.1 Attacks using Future Hallucination

Since the ideal Clairvoyant Attack is impossible, we replace the true future with a 'hallucination' of it. We propose two methods for the hallucination: using a predictive recurrent model (Predictive Attack), and random data substitution (IID Attack).

Algorithm 1 Predictive Attack at time t.

```
count ← 0
while count < MAX_COUNT do
    $\mathcal{L}_{\text{total}}(\delta_t) \leftarrow \mathcal{L}_{\text{adv}}(x_t, y_t^a, \delta_t, h_t^\delta)$
        $+ \sum_{i=t+1}^{t+K} \mathrm{E}_{Q_\phi(x_i|x_{:i-1})}[\mathcal{L}_{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta)]$
        using Monte-Carlo to compute $E_{Q_\phi}[\cdot]$.
    $\forall i \in [t, t+K]$,
    $\delta_i \leftarrow \Pi_{\|\delta_i\|_p \le \epsilon}[\delta_i - \alpha \text{sign}(\nabla_{\delta_i} \mathcal{L}_{\text{total}}(\delta_t)]$
    $\delta_i \leftarrow \text{clip}(x_i + \delta_i) - x_i$
        It forces a valid range of perturbed inputs.
    count ← count + 1
end while
return $\delta_t$
```

Predictive Attack. Predictive Attack (Figure 3-1) uses a future predictive model to mimic Clairvoyant Attack. We define the attack objective of Predictive Attack as follows:

$$\begin{aligned}\delta_t = \arg\min_{\|\delta_t\|_p \le \epsilon} & \underbrace{\mathcal{L}^{\text{adv}}(x_t, y_t^a, \delta_t, h_t^\delta)}_{\text{(A) Current}} \\ & \underbrace{+ \mathrm{E}_{p(x_{t+1:}|x_{:t})}\left[\sum_{i=t+1}^{t+K} \mathcal{L}^{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta)\right]}_{\text{(B) Future, } x_{t+1:} \text{ depends on } x_{:t}.} .\end{aligned} \tag{6}$$

Due to linearity the second term can be simplified as

$$\sum_{i=t+1}^{t+K} \mathrm{E}_{p(x_i|x_{:t})}\left[\mathcal{L}^{\text{adv}}(x_i, y_i^a, \delta_i, h_i^\delta)\right] . \tag{7}$$

Instead of directly modeling the distribution $p(x_{t+1:}|x_{:t})$, we undertake the easier task of generating the future input with a (stochastic) generative model $Q_\phi(x_{t+1}|x_{:t})$ that predicts the next input x_{t+1} given $x_{:t}$ (Figure 3-1, blue). We restrict the number of the prediction steps to K, called **lookahead** since we cannot consider all future inputs with finite resources.

We use another RNN to model the generator Q_ϕ to predict the next input x_{t+1} from $x_{:t}$, using examples of input sequences as a training dataset. (Model details are in Experiments and Appendix B.)

Algorithm 1 describes Predictive Attack's update rule for δ_t, which is a variant of [Mądry *et al.*, 2018]. The hyperparameters MAX_COUNT and α determine the number of updates and the step size of an update. We elaborate more on this in Appendix A.

IID Attack. Hallucinating the future based on an accurate Q_ϕ can be difficult due to the test-time cost of the prediction or the training-time cost of Q_ϕ. To relieve this, we present a heuristic, IID Attack, to replace the prediction model. IID Attack (Figure 3-2) simply ignores the temporal dependence and predicts the future using IID sampling of the input data (Figure 3-2, green), that is, using $E_{p(x_i)}[\cdot]$ instead of $E_{p(x_i|x_{:t})}[\cdot]$ in Equation 6. Practically, this can be done by collecting a sufficient number of past input data and randomly choosing one

Dataset	Task	n	L	Victim Clean Performance
MNIST	C-2	28	28	0.96 (Acc.)
FashionMNIST	C-10	28	28	0.71 (Acc.)
Mortality	C-2	76	48	0.86 (AUC.)
User	C-22	3	50	0.61 (Acc.)
Udacity	R	4096	20	0.05 (MSE)
Energy	R	22	50	0.01 (MSE)

Table 1: Summary of datasets. “C-N” means N-class classification, and “R”means Regression. “n” is a dimension of x_i.

of them as an IID example. Even with the incorrect prediction of IID, it is still using the victim model dynamics (Figure 3-2, red). Such a consideration makes a big difference compared to the current-only greedy perturbation δ_t as we will see.

4.2 Incorporating Different Objectives

The general form of the framework (Equation 1) allows various attacks through the choice of $\mathrm{Agg}(\cdot)$ and the constraint Δ. To showcase our framework’s versatility, we choose γ_i-weighted sum as an instance of $\mathrm{Agg}(\cdot)$:

$$\boldsymbol{\delta} = \underset{\|\delta_i\|_p \le \epsilon,\ \forall i}{\arg\min} \sum_{i=1}^{L} \gamma_i \mathcal{L}^{\text{adv}}(x_i, y \in \{y_i, y_i^a\}, \delta_i, h_i^\delta). \quad (8)$$

In the following, we present three example attacks possible with this aggregation.

Real-time Attack. The real-time attack [Gong *et al.*, 2019] is a special case of this formulation when $\gamma_i = 0$ for $i < L$ and $\gamma_L = 1$, which aims to mislead the last victim output.

Time-window Attack. This attack causes misclassification/prediction at only at specific times interval $[a, b]$. This can be useful when 1) the attack has a more impact at specific times, or 2) the attacker has to avoid detection for a time interval where the victim is vigilant. We can implement this attack by setting $y = y_i^a, \gamma_i = 1$ if $i \in [a, b]$ and $y = y_i, \gamma_i = \tau(> 0)$ otherwise in Equation 8.

Surprise Attack. Surprise Attack induces untargeted error abruptly by maximizing the difference between the maximum error and the mean error over time:

$$\underset{\|\delta_i\|_p \le \epsilon,\ \forall i}{\arg\min} \left[\frac{1}{L} \sum_i \mathcal{L}^{\text{adv}}(x_i, y_i, \delta_i, h_i^\delta) - \max_j \mathcal{L}^{\text{adv}}(x_j, y_j, \delta_j, h_j^\delta) \right]. \quad (9)$$

This attack prevents a victim from reacting properly, thus causing more damage with the same error. For example, an abrupt steering angle change will be more damaging to an autonomous vehicle than a smooth angle change over time.

5 Experiment

We evaluate our attacks to answer the following research questions. **RQ1.** How much does Predictive Attack improve the attack performance?, **RQ2.** How versatile is our online evasion attack framework? **RQ3.** How robust is Predictive Attack?

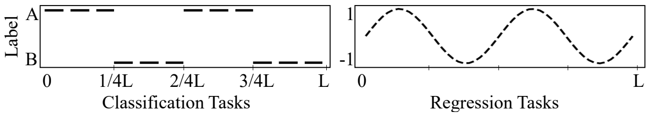


Figure 4: Target label and values of attacks.

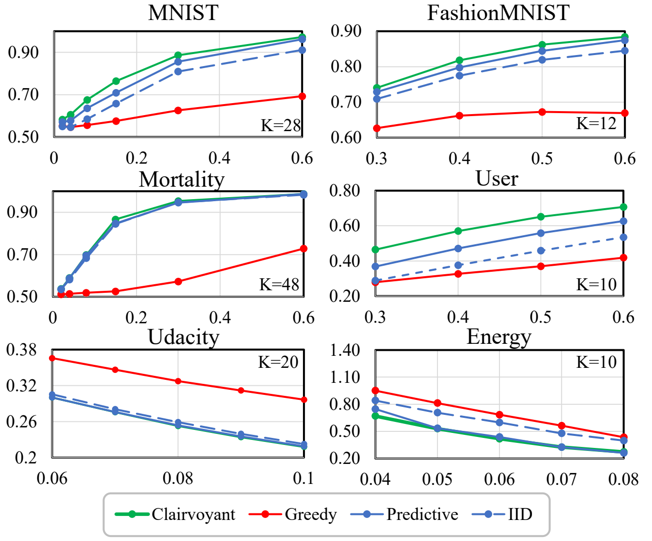


Figure 5: Performance evaluation of Predictive Attack and baselines.

Datasets. We use six datasets for classification and regression in our evaluations as summarized in Table 1.

- **MNIST** [LeCun *et al.*, 1998]: Given a column sequence (vertical lines) of a digit image, predict the correct label of each column. Two classes, 3 and 8, are selected.
- **FashionMNIST** [Xiao *et al.*, 2017]: The same format as MNIST but containing clothing images. We use 10 classes for a harder classification problem.
- **Mortality** [Harutyunyan *et al.*, 2019]: Given a sequential medical record, predict a patient’s mortality every hour.
- **User** [Casale, 2014]: Given a sequence of x-y-z accelerations from a user, predict the user of the sequence.
- **Udacity** [Gonzalez *et al.*, 2017]: Given a sequence of camera images, predict steering angles. We resized the images to 64 x 64 and sequence duration is 0.67s.
- **Energy** [Candanedo *et al.*, 2017]: Given a sequence of 27 weather sensors, predict electricity consumption of a building.

Model Parameters. All models, except for Udacity, consist of one LSTM layer followed by two linear layers with ReLU activations. For Udacity, we use CNN-LSTM as a victim model, and CrevNet [Yu *et al.*, 2020] as Q_ϕ to deal with the high-dimensional images. More model details are in Appendix B.We the Adam optimizer for training with a learning rate of 1e-4. Table 1 summarizes victim’s clean performances. We use ROC-AUC for Mortality to be comparable to the original reports [Harutyunyan *et al.*, 2019].

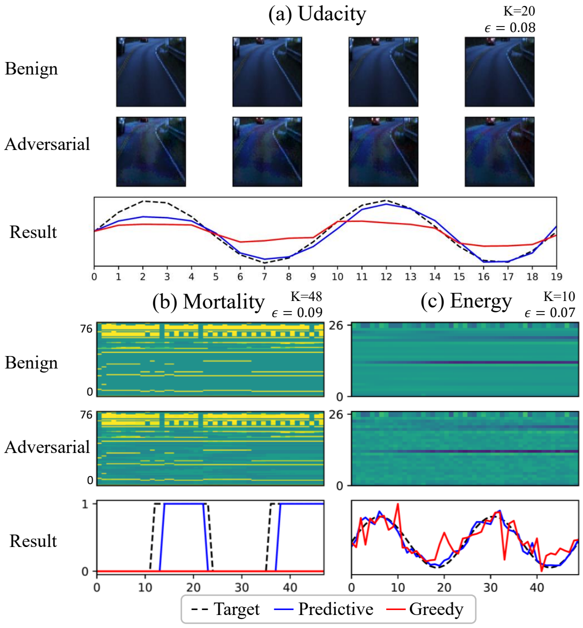


Figure 6: Visualization results of Predictive Attack. We can see Predictive Attack can follow the targeted labels and values much better than Greedy Attack.

Attack Target and Performance Metric. To evaluate the proposed framework and the attack, we use time-varying target outputs for both classification and regression as depicted in Figure 4. It is intended to simulate the dynamic nature of real online attacks better. Appendix F contains more results with other target patterns. An effective attack should achieve high TASR and low TMSE. TASR (Targeted Attack Success Ratio) is the number of time steps where a victim model yields targeted labels, over the total number of time steps L. TMSE (Targeted Mean Squared Error) is the mean squared error between victim model outputs and the targeted values. We use temporal summation as an attack objective (Equation 3) if not specified.

Miscellaneous. The input values range from 0 to 1, and we use ℓ_∞ norm constraints for all tests. We set MAX_ITERS $=$ 100, and $\alpha = 1.5\epsilon/$MAX_ITERS. We report average results of three experiment repetitions retraining a victim model initialized with random weights. Predicted inputs and the perturbed inputs of our attack are presented in Appendices D and E.

5.1 Performance Evaluation

In Figure 5, we answer **RQ1** by comparing the performance of Predictive Attack with Greedy and Clairvoyant. The x-axis is ϵ, ℓ_∞ norm of a perturbation, and the y-axis is the performance metric. On average at maximum ϵ of each plot, the performance of Predictive Attack (straight blue) approaches 98% of Clairvoyant Attacks's TASR (green). Predictive Attack also performs 138% of Greedy Attack's TASR (red). In particular, it is worth noting that safety-critical tasks such as Mortality and Udacity are more prone to attacks than the toy datasets, MNIST and Fashion MNIST.

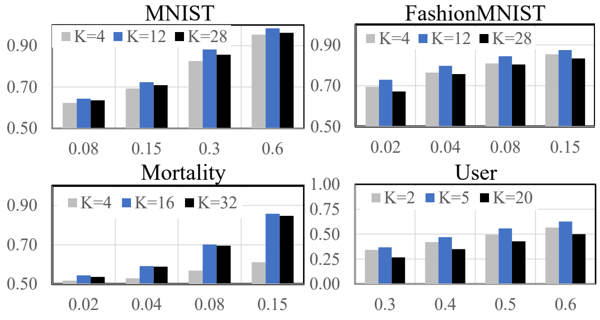


Figure 7: Effect of the lookahead K on Predictive Attack.

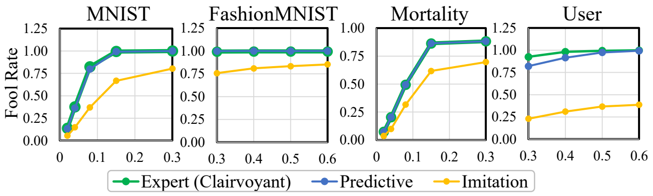


Figure 8: Comparison of attacks with Real-time Attack's objective.

IID Attack's comparable performance (93% of Predictive Attack on average) to that of Predictive Attack shows the importance of victim model dynamics. In particular, in Mortality, IID Attack shows the closest performance to Predictive Attack. We surmise the victim model is more dependent on model dynamics to solve the mortality prediction task. For example, a patient's current severity may depend on a medical record several hours ago, not on the current medical record.

For a qualitative analysis, we visualize the results in Figure 6. We chose (a) Udacity, (b) Mortality, and (c) Energy for visualization. In each figure section, the three rows correspond to benign examples, corresponding adversarial examples, and victim model outputs, respectively. The x-axis is time. We can see that Predictive Attack (blue) closely follows the target values or labels (dashed black) much better than Greedy Attack (red) can.

In Figure 7, we investigate the effect of lookahead K. The x-axis is ϵ, and the y-axis is attack performance. The color of a bar represents a different K. As a result, we find that there is an optimal K. We attribute this to the limitation of Q_ϕ. By increasing K until Q_ϕ can predict accurately, the attack can use the longer temporal dependence and victim model dynamics, leading to the performance improvement. However, if K exceeds the limit, the attack performance decreases because of incorrect future inputs and their wrong perturbations. An attack time should be short to perturb more inputs in a limited time interval. We measured the time per a time step for Predictive Attack to reach 90% of the saturated performance (when MAX_ITERATION is used). For Mortality and Energy, the time is short enough, 0.03 secs and 0.05 secs, considering 3600 secs and 600 secs of each dataset's time step duration. For Udacity, it takes 0.25 secs, which is longer than usual duration of camera input, 0.03 secs. However, we believe that the time can be reduced by using a dedicated hardware or compressing the predictor model.

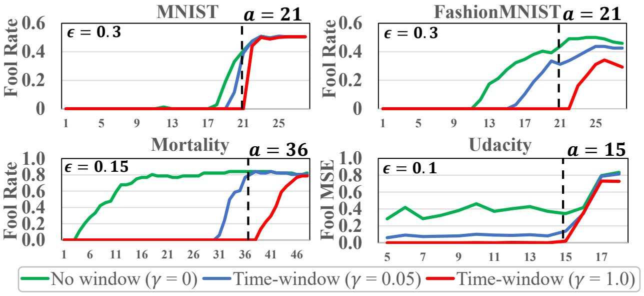


Figure 9: Effect of using the Time-window Attack objective whose purpose is to restrict the error to the interval $[3/4L, L]$. Note that non-window attacks cause error before this interval.

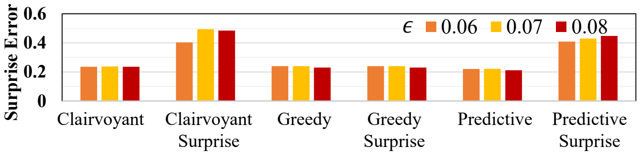


Figure 10: Effect of using the Surprise Attack objective. It aims to cause a sudden error and disrupt the victim from responding properly.

5.2 Versatility of the Attack Framework

Evaluating the effectiveness of different objectives from our framework, we answer **RQ2**, versatility of our framework.

Real-time Attack. In Figure 8, we test a real-time attack objective [Gong *et al.*, 2019] as a special case of our framework. A real-time attacker aims to mislead the last output of a victim model and is untargeted; thus, y-axis is "Fool Rate" that means a frequency of wrong victim decisions. Imitation learning-based real-time attack [Gong *et al.*, 2019] is reimplemented, referring to the public codes[3] (See Appendix K for detail explanations.). Predictive Attack surpasses the imitation learning-based attack (88∼100% vs. 24∼94% of experts' performance). Note that this gain of Predictive Attack comes at the cost of solving PGD, unlike the imitation learning-based attack that depends on a pre-trained agent. Predictive Attack is good for achieving a high attack performance, while imitation learning-based attack is suitable for fast attacks.

Time-window Attack. We demonstrate the attack with temporal specificity. We chose the interval $[a = 3/4L, b = L]$ as the intended window of error. In Figure 9, we present the performance of Predictive Attack with/without the Time-window objective. The x-axis is time. We set the y-axis as "Fool Rate" and "Fool MSE" (MSE between true values and victim outputs) . An ideal attack should cause non-zero values only in $[3/4L, L]$. Predictive Attack fulfills this objective and increases the error after t=a in contrast to non-window attacks. We also find τ controls the trade-off between attack performance and compliance with the time-window.

Surprise Attack. We conduct Surprise Attack experiment with the autonomous driving task from Udacity, where Surprise Attack can be practically important. We define Surprise Error as $\max_i|y_i - f(x_i, h_i^\delta)| - \text{mean}_i|y_i - f(x_i, h_i^\delta)|$. In Figure 10, Predictive Attack with Surprise objectives achieves about 2.09 times higher Surprise Error than a naive Predictive Attack and Greedy at $\epsilon = 0.08$.

Dataset	ϵ	K	Predictive $\eta = 0$	Predictive $\eta = 0.4$	Greedy
MNIST	0.08	8	0.66	0.64	0.56
FashionMNIST	0.30	8	0.76	0.74	0.63
Mortality	0.15	32	0.85	0.80	0.52
User	0.30	10	0.34	0.25	0.28
Udacity (MSE)	0.05	16	0.35	0.37	0.41

Table 2: Predictive Attack against incorrect future prediction.

Dataset	ϵ	K	Whitebox	Graybox
MNIST	0.3	28	0.86	0.64
FashionMNIST	0.5	28	0.88	0.47
Mortality	0.15	32	0.85	0.75
User	0.5	10	0.54	0.21
Udacity (MSE)	0.06	16	0.30	0.47

Table 3: Predictive Attack when model parameters are unknown.

5.3 Robustness Evaluation

To answer **RQ3**, we evaluate Predictive Attack under a variety of unseen situations in our attack framework.

Incorrect Future Prediction. We investigate the performance of Predictive Attack under degraded Q_ϕ. To control the prediction quality, we replace a predicted future input x_t with $x_i^\eta = (1-\eta)x_i + \eta e$, where e is a uniform random variable in the valid input range. In Table 2, although slightly decreased as the noise is added ($\eta = 0.4$), Predictive Attack performs better than Greedy Attack, using the victim model dynamic as consist with the case of IID Attack in Figure 5.

Unknown Model Parameters. To evaluate the robustness under limited victim information, a transfer attack [Liu *et al.*, 2017] is conducted (Table 3). We assume a gray-box threat model where an attacker knows a victim model's architecture but not model parameters. Adversarial examples generated from an attacker-trained surrogate model are transferred to the actual victim model. Transfer attack achieves average 63% performance of white-box attack in the classification tasks, up to 88% in Mortality.

6 Conclusion

This paper introduces a general framework for online evasion attacks on recurrent models. Our framework can accommodate various time-varying attack objectives and constraints, allowing a comprehensive robustness analysis. Based on our framework, we propose Predictive Attack and IID Attack. The success of these attacks highlights the new surface of attack for recurrent models, which need to be addressed. However, defense in the online setting has not been fully studied yet, while existing offline defenses [Mądry *et al.*, 2018; Zhang *et al.*, 2019] are not suitable for online tasks. We leave it as future work to investigate online defense methods.

[3] https://github.com/YuanGongND/realtime-adversarial-attack

Acknowledgements

This work was supported in part by ERC (NRF-2018R1A5A1059921) and IITP (2020-0-00209) funded by the Korea government(MSIT). It is also supported in part by the NSF EPSCoR-Louisiana Materials Design Alliance (LAMDA) program #OIA-1946231.

References

[Candanedo *et al.*, 2017] Luis Miguel Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017.

[Casale, 2014] Pierluigi Casale. User Identification From Walking Activity. UCI Machine Learning Repository, 2014.

[Croce and Hein, 2020] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, 2020.

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

[Fawaz *et al.*, 2019] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Adversarial attacks on deep neural networks for time series classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[Gondim-ribeiro *et al.*, 2018] George Gondim-ribeiro, Pedro Tabacof, and Eduardo Valle. Adversarial Attacks on Variational Autoencoders. *arXiv:1806.04646*, 2018.

[Gong *et al.*, 2019] Yuan Gong, Boyang Li, Christian Poellabauer, and Yiyu Shi. Real-time adversarial attacks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 4672–4680, 2019.

[Gonzalez *et al.*, 2017] Eric Gonzalez, MacCallister Higgins, and Oliver Cameron. Udacity self-driving car challenge. https://github.com/udacity/self-driving-car, 2017. Accessed: 2022-06-05.

[Goodfellow *et al.*, 2015] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[Harutyunyan *et al.*, 2019] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

[Huang *et al.*, 2020] Wenhui Huang, Jason Gu, Xin Ma, and Yibin Li. End-to-end multitask siamese network with residual hierarchical attention for real-time object tracking. *Applied Intelligence*, 50(6):1908–1921, 2020.

[Kiran *et al.*, 2021] Bangalore Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE, 86(11):2278–2324*, 1998.

[Liu *et al.*, 2017] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations (ICLR)*, 2017.

[Mądry *et al.*, 2018] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[Oregi *et al.*, 2018] Izaskun Oregi, Javier Del Ser, Aritz Perez, and Jose A Lozano. Adversarial sample crafting for time series classification with elastic similarity measures. In *International Symposium on Intelligent and Distributed Computing*, pages 26–39. Springer, 2018.

[Pinto *et al.*, 2021] Tiago Pinto, Isabel Praça, Zita Vale, and Jose Silva. Ensemble learning for electricity consumption forecasting in office buildings. *Neurocomputing*, 423:747–755, 2021.

[Suradhaniwar *et al.*, 2021] Saurabh Suradhaniwar, Soumyashree Kar, Surya S Durbha, and Adinarayana Jagarlapudi. Time series forecasting of univariate agrometeorological data: A comparative performance evaluation via one-step and multi-step ahead forecasting strategies. *Sensors*, 21(7):2430, 2021.

[Szegedy *et al.*, 2014] Christian Szegedy, Joan Bruna, Dumitru Erhan, and Ian Goodfellow. Intriguing properties of neural networks. In *In International Conference on Learning Representations (ICLR)*, 2014.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv, cs.LG/1708.07747*, 2017.

[Xie *et al.*, 2020] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1738–1742. IEEE, 2020.

[Yu *et al.*, 2020] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2020.

[Zhang *et al.*, 2019] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.