Adversarial Attacks on Deep Temporal Point Process

1st Samira Khorshidi
Computer and Information Science
Indiana University-Purdue University Indianapolis
Indianapolis, USA
sakhors@iupui.edu

2nd Bao Wang
Department of Mathematics
University of Utah
Salt Lake City, USA
wangbaonj@gmail.com

3rd George Mohler
Department of Computer Science
Boston College
Boston, USA
mohlerg@bc.edu

***Abstract*—Temporal point processes have many applications, from crime forecasting to modeling earthquake aftershocks sequences. Due to the flexibility and expressiveness of deep learning, neural network-based approaches have recently shown promise for modeling point process intensities. However, there is a lack of research on the robustness of such models in regards to adversarial attacks and natural shocks to systems. Precisely, while neural point processes may outperform simpler parametric models on in-sample tests, how these models perform when encountering adversarial examples or sharp non-stationary trends remains unknown. Current work proposes several white-box and black-box adversarial attacks against temporal point processes modeled by deep neural networks. Extensive experiments confirm that predictive performance and parametric modeling of neural point processes are vulnerable to adversarial attacks. Additionally, we evaluate the vulnerability and performance of these models in the presence of non-stationary abrupt changes, using the crimes dataset, during the Covid-19 pandemic, as an example.**

***Index Terms*—Point process, Adversarial attacks, Deep learning, Nonparametric modeling**

I. Introduction

Temporal point processes utilized in sensitive fields and security-related tasks such as analyzing electronic transaction records [1], forecasting earthquake aftershocks [2], mitigating the spread of fake news [3], and allocating police to crime hot spots [4]. Considering the deep learning approaches' high success rate and influence in various domains, neural network-based approaches to modeling point processes have recently received attention from the research community [5]–[7] to capture real event patterns better than parametric models. However, deep learning approaches risk over-parameterizing models and overfitting real-world, noisy data despite their success. Furthermore, there is a lack of research into how robust such models are to natural shocks to systems, e.g., pandemic and adversarial attacks impact deep point process forecasts of crime. Moreover, adversarial samples can be transferable from one deep learning network to another and become a black-box attack which is a more elevated risk. Such research is crucial for security purposes and understanding deep learning models to make machine learning models trustworthy.

A. Contribution

Despite the remarkable success of deep neural networks (DNN)s, they suffer from severe vulnerabilities to adversarial attacks. Vulnerability's examination of DNN in computer vision and natural language processing has received attention recently. However, to the best of our knowledge, we are the first to explore the adversarial methods for point processes modeled by DNN and examine their performance. In particular, I) We propose several adversarial methods to generate white-box and black-box adversarial attacks on point processes modeled by DNN. II) We show how adversarial attacks can disturb underlying parameters of point processes which are considered a threat to parametric modeling. Furthermore, III) We illustrate how susceptible deep point processes are to natural shocks and non-stationary changes in data.

B. Organization

The rest of the paper is organized as follows: Section II revisits some basic definitions in point processes and discuss the related work in recurrent neural networks (RNNs), adversarial attacks of deep learning models, and the robustness of such models. Section III sets up the problem and formalizes the algorithm for crafting adversarial examples. Section IV discusses the utilized datasets. Section V presents our results and discusses various attacks by comparing the models performance, and the last section concludes the paper.

II. Related work

Temporal point processes are practical mathematical tools for modeling event data in which the inter-event times as a random variable are modeled. Therefore, there is no required time window to aggregate events, which may cause discretization errors; this is the main difference between point process models and the discrete-time representation utilized in time series analysis [8]. Moreover, point processes can be deterministic or stochastic, and non-stationary Poisson, self-correcting, and Hawkes process are stochastic point processes that we have utilized in the current work.

As a result of advances in deep learning techniques, researchers have proposed RNN to model the intensity function of point processes [5], [9]. Most of the proposed methods uti-

lize Long-Short Term Memory (LSTM) [10]; that we reiterate its formulation as follows

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1} + \mathbf{b}_f),\\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t + \mathbf{b}_o),\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{1}$$

where $\odot$ is element-wise multiplication, and σ is the logistic sigmoid function. The above system can be abstract as an LSTM equation as $(\mathbf{h}_t, \mathbf{c}_t) = \mathrm{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1} + \mathbf{c}_{t-1})$ [5].

Such a model can encode a nonlinear link between the predicted transient occurrence intensity of different types of events with the history of participator events, which is more expressive, and it can model more complexity in data than previous parametric or non-parametric models [11].

The RNN-based model proposed in [5] provides a black-box approach to model the intensity while both background and effect of history are considered. In their modeling, presented in (2) below, the background intensity is modeled by an RNN as a time series with its units aligned with time series indexes, $\{y_t\}_{t=1}^{T}$, while another RNN handles the historical events whose units are aligned with asynchronous events to capture the long-range dynamics, $\{z_i, t_i\}_{t=1}^{N}$.

$$\begin{aligned}
(\mathbf{h}_t^y, \mathbf{c}_t^y) &= \mathrm{LSTM}_y(\mathbf{y}_t, \mathbf{h}_{t-1}^y + \mathbf{c}_{t-1}^y),\\
(\mathbf{h}_t^z, \mathbf{c}_t^z) &= \mathrm{LSTM}_z(\mathbf{z}_t, \mathbf{h}_{t-1}^z + \mathbf{c}_{t-1}^z),\\
\mathbf{e}_t &= \tanh(\mathbf{W}_f[\mathbf{h}_t^y, \mathbf{h}_t^z] + \mathbf{b}_f),\\
\mathbf{U}_t &= \mathrm{softmax}(\mathbf{W}_U\mathbf{e}_t + \mathbf{b}_U),\\
\mathbf{u}_t &= \mathrm{softmax}(\mathbf{W}_u[\mathbf{e}_t, \mathbf{U}_t] + \mathbf{b}_u),\\
s_t &= \mathbf{W}_s\mathbf{e}_t + b_s,
\end{aligned} \tag{2}$$

In (2), the subscripts U and u indicate the primary- and secondary-type of events, and s is the timestamp of each event.

Besides achievements in neural networks and their applications, recent research has revealed that neural network models in practice are vulnerable to misclassifying adversarial samples that have been crafted by adding an imperceptible additive perturbation to the data samples. In deep learning models, such a vulnerability was assumed to be explained by nonlinearity and overfitting. However, [12] argues such an assumption and shows deep learning models are vulnerable because of the linearity of adversarial perturbations, which can be analyzed as a property of high-dimensional dot products. On the other hand, neural networks, e.g., ReLUs, and LSTMs networks, are overly linear to oppose linear adversarial perturbation. In their work, they have suggested fast methods to generate adversarial samples, such as the fast gradient sign method (FGSM) [12], and claimed in adversarial sample creation, the perturbation's direction is more important than the specific point in space.

In addition to security concerns, research in the robustness of deep learning models are showing study of adversarial examples crafted under limited situations is helpful since it provides new insight into the geometrical characteristics and behavior of models in high-dimensional space; e.g., the characteristics of adversarial images close to the decision boundaries can help describe the boundaries' shape [13].

Recent research experiments have shown the amount of perturbation to fool deep learning models can be extremely limited, as [14] proposed a low cost, black-box attack to fool visionary deep learning models, where the only available information is the probability labels and only one pixel can be modified based on differential evolution (DE). One critical property of such attacks is their flexibility; they can attack different networks regardless of their differentiability.

Generally, adversarial attacks are not limited to visionary, and speech-to-text systems are also exposed to misclassifying adversarial samples. Reference [15] examines the adversarial attacks in the audio domain using the Connectionist Temporal Classification Loss Function (CTCLF) as an attack mechanism and PCA as an attack and defense mechanism. In this experiment, CTCLF and PCA, as black-boxed approaches, have successfully attacked DeepSpeech[1]. In contrast, PCA as a defense mechanism does not improve the performance of DeepSpeech against adversarial attacks.

Point processes and viral processes are confirmed to be sensitive to changes in network structure. Reference [16] has shown the evolution of viral processes on a network is highly sensitive to the structural features of the network. They have discussed that assortativity and degree distribution cannot fully explain the variance in the spread of viruses; instead, graphlet distribution can explain such a variance.

Additionally, Hawkes process is used in anomaly detection. For example, reference [17] proposes a framework using the multivariate Hawkes process and reinforcement learning as a fake news mitigation framework on networks. The point process defines "mitigation" in this work on the network, and finding the optimal mitigation strategy is the objective that determines how to adjust the exogenous intensity of the few mitigator nodes on the network.

Adversarial attacks on time series models are another related work. In [18], the authors propose adversarial attacks on deep learning time series classifiers using the fast gradient sign method (FGSM), and the basic iterative method (BIM) [19], [20]. However, their methods are considered black-box since adversarial samples are crafted using the gradient of ResNet [21], rather than the targeted network. Lastly, [22] proposes adversarial attacks on Convolutional Neural Network (CNN), LSTM, and Gated Recurrent Unit (GRU) networks as multivariate time series regressions where the adversarial samples are crafted using FGSM and BIM.

III. Methodology

The robustness of neural point processes to natural shocks and adversarial attacks remains an open problem to date. We leverage existing research on adversarial attacks in vision and signal processing and extend such methods, when possible, to temporal point processes. We additionally examine the transferability of attacks and compare their performance in

[1]Speech-to-text neural network implemented by Mozilla

transferred black-box settings. We especially believe methods developed for time series data will be applicable to point processes since they share common characteristics such as sequential and noisy data; the difference is that point processes model discrete events using a continuous intensity and time series bin events or other variables in time.

A. Models

This work is limited to three type of point processes, non-stationary Poisson, self-correcting, and Hawkes processes, where the main focus is Hawkes processes. In non-stationary Poisson process, unlike regular Poisson process, the average rate of events is allowed to change by time. Non-stationary Poisson has all properties of a Poisson process, except for the fact that the intensity is a function of time, i.e. $\lambda = \lambda(t)$, instead of being fixed.

A point process N is called self-correcting if $cov(N(s,t), N(t,u)) < 0$ for $s < t < u$. In this formulation, cov denotes the co-variance of the two quantities [23]. Intuitively, due to the negative correlation, past points' occurrence, inhibits the future points' occurrence [24].

Lastly, in Hawkes processes [25], [26], the event rate is not fixed, but is dependent on some random inputs, including the history of the process. Hawkes process is a self-exciting process, each arrival increases the rate of future arrivals for some time and is determined by a background Poisson process $\lambda_0(t)$, which reflects spontaneous events and at each event in the history a Poisson process g is centered at that event reflecting the increase in the intensity in near future. In summary, the intensity of the Hawkes process can be modeled as follows

$$\lambda(t) = \lambda_0(t) + \sum_{t_i<t} g(t - t_i), \tag{3}$$

where $\lambda(t)$ donates the event rate at time t.

In this work, we explore the performance of two deep neural networks-base point processes. For the exponential hazard (EXP) model as proposed by the authors of [9] and followed by [6], the inter-event time, $x_i = (t_i - t_{i-1})$ is fed into a the RNN and the hidden unit of RNN is updated by $h_i = f(W^h h_{i-1} + W^x x_i + b_h)$. Here f represents the activation function, and W^h, W^x, and b_h are the recurrent weight matrix, input weight matrix, and bias term, respectively [9]. The conditional intensity is a function of the elapsed time from the latest event and the hidden state of the RNN, as: $\lambda(t|H_t) = \phi(t - t_i|\mathbf{h}_i)$, and ϕ is a non-negative function that is the hazard function with the following form as assumed by [9]:

$$\phi(\tau|h_i) = \exp(w_t\tau + v^\phi.\mathbf{h}_i + b^\phi) \tag{4}$$

$\tau_i = t_{i+1} - t_i$ is the inter-event interval.

The other examined model is the fully neural network-based (NN) model for general temporal point process [27], that relaxes the constraints on the time course for hazard functions of point processes while they are modeled using RNNs. In this model, the cumulative intensity function is modeled by the integral of intensity function and the instantaneous intensity is obtained by taking the derivative of the cumulative intensity function. Such a model allows us to have flexible and general intensity function with exact evaluation. Formally, instead of modeling the hazard function ϕ, in this model the cumulative hazard function $\Phi(\tau, \mathbf{h}_i)$ is modeled where;

$$\Phi(\tau, \mathbf{h}_i) = \int_0^\tau \phi(s, \mathbf{h}_i)ds \tag{5}$$

And, one can achieve the hazard function by;

$$\phi(\tau, \mathbf{h}_i) = \frac{\partial\Phi(\tau, \mathbf{h}_i)}{\partial\tau} \tag{6}$$

In this setting, $\mathbf{h}_i$ is the hidden state of the RNN and $\tau_i = t_{i+1} - t_i$ is the inter-event interval [27].

B. Adversarial attacks

In this section, we present adversarial attacks for point process models. In regression problems, adversarial attacks can be defined based on numerical instability of the models. The numerical (in-)stability of an algorithm is defined based on the extent to which a function's output changes with changes in the input [28] and adversarial attacks are toward increasing the instability of the model where defense mechanisms decrease it. Formally, considering a neural regression model $T(x, \theta)$, where $T : \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_m}$ has N_0 scalar inputs and N_m scalar outputs, the objective of $T(x, \theta)$ for $x \in \mathbb{R}^{N_0}$ and corresponding target $y \in \mathbb{R}^{N_m}$ is [29]

$$\underset{\theta}{\operatorname{argmin}} l(T(x, \theta), y) \tag{7}$$

Then given $T(x, \theta)$, in the adversarial attack setting, the objective of adversary with budget ϵ is to maximize the instability of $T(x, \theta)$, which is mathematically formulated below

$$\underset{||x_{adv}-x||_p\leq\epsilon}{\operatorname{argmax}} \; l(T(x_{adv}, \theta), y) \tag{8}$$

where $p = 1, 2$, or ∞. Notice that the optimal solution, x_{adv}, to the above optimization problem is not unique.

Furthermore, depending on the algorithm utilized in adversarial attack generation and the degree of provided information to the attacker, attacks can be white-box or black-box attacks. For white-box attacks, the attacker is fully aware of the internals of the target model and its weights (θ) and uses the model's gradient to find the vulnerable regions of the input space that affects the model's output drastically. In a black-box setting, no information about the attacked network is provided to the attacker. However, the attacker can compose transferable adversarial perturbations to the target model using an alternate model. In existing work, we have crafted the adversarial samples by strategies presented in the following to attack temporal point processes.

1) Fast Gradient Sign Method (FGSM) : FGSM [12] is a bounded attack initially proposed for the visionary. Here, we extend it to point process regression, and the adversarial sample is formed by using perturbations in the direction of the

gradient. The adversarial sample generated by the FGSM can be formulated as:

$$X^{adv} = clip(X + \epsilon \cdot sign(\nabla_x \mathcal{J}(\theta, X)), min(X), max(X)) \tag{9}$$

where $\mathcal{J}(\theta, X))$ is the required training's cost of the neural network, ϵ is the perturbation factor, and $clip(\cdot, a, b)$ function squeezes its input to the range of $[a, b]$.

2) Iterative Fast Gradient Sign Method (iFGSM): ifGSM, also known as Basic Iterative Method (BIM) [19], [20], is an iterative form of FGSM that, instead of one single step, k steps attack with a budget α is applied iteratively as presented in (10).

$$\begin{gathered} X^{adv0} = X \\ X^{adv_{t+1}} = X^{adv_t} + \alpha \cdot sign(\nabla_x \mathcal{J}(\theta, X^{adv_t})) \\ X^{adv_{t+1}} = clip(X^{adv_{t+1}}, X^{adv_{t+1}} - \epsilon, X^{adv_{t+1}} + \epsilon) \\ X^{adv} = X^{adv_k} \end{gathered} \tag{10}$$

3) Projected Gradient Descent (PGD): PGD attack as presented by [20] without random start, is a more potent iterative version of FGSM. Here, we apply PGD as a generalization of iFGSM with random initialization.

$$\begin{gathered} X^{adv0} = X + \textbf{clip}(\mathcal{N}(0^d, I^d), -\epsilon, +\epsilon) \\ X^{adv_{t+1}} = X^{adv_t} + \alpha \cdot sign(\nabla_x \mathcal{J}(\theta, X^{adv_t})) \\ X^{adv_{t+1}} = \textbf{clip}(X^{adv_{t+1}}, X^{adv_{t+1}} - \epsilon, X^{adv_{t+1}} + \epsilon) \\ X^{adv} = X^{adv_k} \end{gathered} \tag{11}$$

4) Momentum Iterative Fast Gradient Sign Method (miFGSM): miFGSM [30] is a transformation of iFGSM such that, before applying FGSM with a budget α, the gradient of the previous t steps with a decay factor μ is employed to update the gradient at step $t+1$. In this approach, the update directions are stabilized, and the algorithm skips poor regional maxima during the iterations. Therefore, the crafted adversarial samples are more transferable. However, despite the high cost of miFGSM attack, as we see in section V, it is not generating more transferable samples in comparison to previous single step and iterative attacks.

$$\begin{gathered} X^{adv0} = X, g_0 = 0 \\ g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{J}(\theta, X^{adv_t})}{|\nabla_x \mathcal{J}(\theta, X^{adv_t})|_1} \\ X^{adv_{t+1}} = X^{adv_t} + \alpha \cdot sign(g_{t+1}) \\ X^{adv_{t+1}} = clip(X^{adv_{t+1}} X^{adv_{t+1}} - \epsilon, X^{adv_{t+1}} + \epsilon) \\ X^{adv} = X^{adv_k} \end{gathered} \tag{12}$$

5) RAND+FGSM (R+FGSM): R+FGSM has been proposed by [31] to attack the adversarially trained neural networks. Here, in one step, a small random perturbation with size α is applied to the input before applying FGSM of $(\epsilon - \alpha)$ cost. R+FGSM is a randomized, single-step and computationally efficient form of PGD.

6) Saliency Map (SM): A saliency map in computer vision indicates the level of significance of a pixel to the human visual system that has application in region-of-interest extraction, image cropping, image captioning, and beyond [32], [33]. Similarly, we propose another single step, low cost adversarial attack where we first identify the *important* events, depending on the gradient of the neural network at a particular event, within the input sequence and then perturb identified events by the adversarial saliency map approach to achieve the adversary sample. Formally, we propose the following attack strategy

$$\begin{gathered} X^{adv} = X \\ X_i^{adv} = X_i + \epsilon \cdot sign(\nabla_x \mathcal{J}(\theta, X_i)) \\ X_j^{adv} = X_j - \epsilon \cdot sign(\nabla_x \mathcal{J}(\theta, X_j)) \end{gathered} \tag{13}$$

where event i maximizes $\nabla_x \mathcal{J}(\theta, X)$ and event j minimizes it w.r.t. the input event sequence.

7) Gradient descent (GD): In the proposed gradient descent adversarial attack, an adversarial event time within interval of first and last event of the sequence is added to the sequence, using the procedure presented in Algorithm 1.

Algorithm 1 Gradient descent adversarial attack procedure

procedure GDADVERSARIAL(X, ϵ, α)
  $X^{adv} \leftarrow X$ ▷ Initialization
  $event \leftarrow$ genRandomVal$[\min(X^{adv}), \max(X^{adv})]$
  $idx \leftarrow$ genRandomInt$[0, length(X^{adv})]$
  X^{adv}.*pop*(0)
  X^{adv}.insert$(idx, event)$
  while $(\nabla_x \mathcal{J}(\theta, X_{idx}^{adv})$ is changing) and $\alpha < \epsilon$ **do**
    $event \leftarrow event + (\alpha \cdot \nabla_x \mathcal{J}(\theta, X_{idx}^{adv}))$
    $X^{adv}[idx] \leftarrow event$
    $idx \leftarrow getIdx(X^{adv}, event)$
  end while
  return X^{adv}
end procedure

8) Time manipulation (TM): In point process applications, e.g., crime forecasting based on reported crimes to the police departments, one can easily report a fake crime to mislead the point process algorithm. Following the same context, we propose a time manipulation attack. Here, we remove the first event in the event sequence and add a random event within the interval of the first and last event's time regardless of the model weight and architecture. Therefore, we consider TM as a black-box version of the GD attack since it is independent of the network architecture and the loss function.

C. Transferability and transferable black-box setting

Regardless of the adversarial attacks' strength against deep learning models, the transferability of adversarial samples has raised concerns in literature [12], [34], [35] where adversarial samples between two independent trained models are transferred. In the current work, we study this phenomenon by proposing transferable adversarial attacks, i.e., we craft the adversarial instances using the presented white-box attacks on

a model with similar architecture and more trainable parameters. Then, the adversarial samples are employed to attack the target model. Attackers can have more neural network layers, RNN units, or hidden neural network units such that,

- RNN units: Means the number of units in an RNN layer.
- Network depth (layers): Means the number of hidden layers of the cumulative hazard function network.
- Hidden neural network (NN) units: The number of units in each hidden layer of NN model [27].

D. Baseline and Metrics

Baseline corresponds to the standard case when a model is built without any adversarial assumptions. We will compare each potential adversarial attack's performance when under attack to the baseline and its performance when no attack has occurred. Our metrics in this comparison are:

- *Mean Negative Log Likelihood*
- *Mean Absolute Error*
- *Fooling Error*

$$FE = \frac{1}{K}\sum_{k=1}^{K} ||T(x_k^{adv}) - T(x_k)||_q \tag{14}$$

- *Symmetric Mean Accuracy Percentage Error [29]*

$$SMAPE = \frac{2}{K_+}\sum_{k=1}^{K_+} \frac{||T(x_k^{adv}) - y_k||_q - ||T(x_k) - y_k||_q}{||T(x_k^{adv}) - y_k||_q + ||T(x_k) - y_k||_q} \tag{15}$$

Where q norm in FE and SMAPE metrics, must match the l_p norm employed in generating adversarial attacks and SMAPE is limited to the K_+ positive elements in the summation.

IV. Data

Similar to [9], [27], experiments are run on the following point process datasets. We split each dataset into train and test sets. The train set is then used to estimate the model's parameters in the training phase, and the model's performance on the test set is quantified by the evaluation metrics.

- ***Hawkes Process*** In this work, we have simulated [26] 100,000 event times from the Hawkes process with the sum of exponentials kernel, and the conditional intensity function is given by:

$$\lambda(t|H_t) = \mu + \Sigma_{t_i<t}\Sigma_j^M \alpha_j\beta_j \exp\{\beta_j(t-t_i)\}, \tag{16}$$

  where $M = 1$, $\mu = 0.2$, $\alpha_1 = 0.8$, and $\beta_1 = 1.0$. The ratio $\frac{\alpha}{\beta}$ is known as *branching ratio* and is the declaration of the Hawkes process regime and if the Hawkes process explodes (super-critical regime and $\frac{\alpha}{\beta} > 1$), or not [26].
- ***Self-correcting Process*** For Self-correcting process, 100,000 event times are simulated by the conditional intensity function that is given as $\lambda(t|H_t) = \exp(t - \sum_{t_i<t} 1)$.
- ***Non-stationary Poisson Process (N-Poisson)*** Finally for non-stationary Poisson dataset, 100,000 events time are simulated from the following conditional intensity function $\lambda(t|H_t) = 0.99\sin(\frac{2\pi t}{20000}) + 1$ as suggested by [27].
- ***Crimes and Covid-19*** To assess the performance of models concerning non-stationary changes, such as the effect of the Covid-19 pandemic on reported crimes in Chicago, we use the reported crimes to the Chicago police department from 2016 to 2018 as training, and crimes in 2019 and 2020 as test set, separately[2].

V. Results and Discussion

This section presents the results of our adversarial attacks on the deep point process models on the specified datasets. We present the predictive performance-related experiments on all point process datasets. In adversarial attacks on regression models, we expect some changes in response for any changes in the input. In this situation, the adversary aims for a dramatic change in output for a small change in input [28]. According to the experiments, to have high effect, the perturbation factor (ϵ) needs to be large enough, although with $\epsilon = 0.1$, the change in performance is statistically significant yet. In all of our attacks, ϵ is set to 0.1, and for iterative attacks, $k = 10$ and $\alpha = \epsilon/10$.

We see that the choice of metric is a critical factor in evaluating adversarial attacks' effectiveness. Specifically, we illustrate that MAE is the least expressive evaluation metric in an adversarial setting. Furthermore, through extensive experiments, we see our proposed PGD can be considered as a "universal first-order attack" such that by first-order, we mean the adversarial attacks solely depend on the gradient of the neural network as suggested by [19]. However, our results reveal that such a generalization is limited to first-order adversarial attacks utilizing the entire gradient vector.

From the results, iFGSM has the best overall performance in white-box attacks against both fully neural network (NN) and exponential kernel (EXP) models. Furthermore, similar to [19], our results suggest that the transferability of attacks decreases as the power of attack increases, and single-step attacks have more transferability in comparison to iterative attacks.

We discuss adversarial attacks manipulating the parameters and branching ratio of the Hawkes process. Finally, we investigate the effect of non-stationary abrupt changes on the models' performance, using the crimes dataset during the Covid-19 era.

In Fig. 1, we present two adversarial samples along with the conditional intensity values for each sequence and simulation of the model prediction for the real and adversarial input. From the examples, it can be inferred that unlike their intensity, the adversarial samples are close to the real sample in time space. In Table I, the performance of the fully neural network model on the specified datasets as well as the most effective white-box and black-box attacks are presented. Consecutively, in Table II, we present the performance of exponential hazard model on the same datasets and attacks.

From Table I and Table II, against the EXP model, iFGSM is not as effective as the NN model, but it still increases the predictive uncertainty (MNLL).

In Fig. 2, we compare all attacks for both the NN and the EXP models on Hawkes process dataset. As illustrated, with

[2]https://data.cityofchicago.org/Public-Safety/Crimes-Map

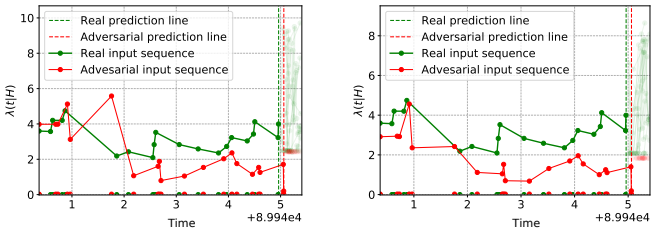

Fig. 1: Real input sequence (Green) and the corresponded adversarial sequence (Red) on Hawkes data. Left: Transferred FGSM from larger network, layers = 4. Right: Transferred FGSM from larger network, RNN=256. The vertical axis represents the conditional intensity.

TABLE I: Predictive performance of fully neural network (NN) model in presence of the most effective white-box and black-box attacks.

Point process	**Attack**	**MNLL** (e-01)	**MAE** (e-01)	**FE** (e-02)	**SMAPE** (e-01)
Hawkes	No Attack	4.1	7.77	NA	NA
	iFGSM	6.54	8.21	4.75	3.66
	TM	5.98	8.36	2.12	7.08
NS-Poisson	No Attack	9.83	10.0	NA	NA
	iFGSM	10.41	10.38	4.64	1.573
	TM	12.91	11.46	40.56	4.05
Self-correcting	No Attack	8.21	4.97	NA	NA
	iFGSM	16.31	9.52	48.00	7.77
	TM	11.94	7.05	49.79	8.32

$*$ We recognize a white-box attack as "effective", if both FE and SMAPE are the highest among first-order white-box attacks.
The column-wise scale of each metric is presented in () below it.

respect to Fooling error (FE) and Symmetric Mean Accuracy Percentage Error (SMAPE), TM is the most effective attack, although, unlike others, TM is a single-step, single-point, black-box attack, and the perturbation is local. Conversely, in first-order attacks, the perturbation is dispersed over the input sample, and the attack affects every event time in the sequence utilizing the gradient of the network. Such outcomes imply despite the history term in the Hawkes process's conditional intensity, the model is vulnerable to spikes in the input and more robust against global perturbations.

In adversarial attacks on neural point processes, the models'

TABLE II: Predictive performance of exponential hazard (EXP) model in presence of the most effective attacks.

Point process	**Attack**	**MNLL** (e-01)	**MAE** (e-01)	**FE** (e-02)	**SMAPE** (e-01)
Hawkes	No Attack	4.63	8.01	NA	NA
	iFGSM	5.67	8.26	3.66	2.32
	TM	6.67	8.72	28.58	6.27
NS-Poisson	No Attack	9.7	10.2	NA	NA
	miFGSM	10.32	10.60	5.27	1.65
	TM	13.36	11.42	37.74	4.25
Self-correcting	No Attack	7.82	4.94	NA	NA
	iFGSM	18.86	9.65	50.41	8.01
	TM	11.49	6.91	49.02	8.36

Fig. 2: Effect of adversarial attacks against NN (Blue) and EXP (Orange) models on their predictive performance.

predictive performance is the goal of the attacks. Not surprisingly, the estimated parameters are also vulnerable if the neural point process model is used to estimate the intensity function. Considering an application of the Hawkes process, e.g., crime forecasting, the intensity of the event's arrival is increased by an event occurrence and then exponentially decays towards the baseline level. Therefore, the values of μ, α, and β influence the model's behavior in predicting the arrival time of the event.

In the studied Hawkes process in the current work, since the background intensity, μ, is fixed, depending on the branching ratio, the Hawkes process will explode and will be in super-critical regime, if $\frac{\alpha}{\beta} > 1$, as we discussed in section IV.

From Table III, the test set is a subcritical ($\frac{\alpha}{\beta} < 1$) Hawkes process. Moreover, on adversarial sets, for both the NN and EXP models, all adversarial sets, except GD against NN, are subcritical Hawkes processes. However, the predicted events by the NN model are all super-critical Hawkes processes, except for predictions in presence of SM and GD attacks. On the other hand, the predictions made by the EXP model are all subcritical Hawkes processes, except for GD. Preserving the Hawkes process's regime is an important characteristic and results suggest the NN model cannot maintain it.

From here, we recommend considering the effects of adversarial attacks on the predictive performance and point process parameters to be regarded as *Micro* and *Macro* effects, respectively. In the Micro effect, the effect of an adversarial attack is limited to each input sequence and it's reflected in predictive performance. In contrast, for the Macro effect, the adversarial attack may affect the modeled point process parameters and this affect the parametric modeling performance. Presented results both on predictive performance and Hawkes process parameters suggest that although adversarial attacks show disagreement in their Micro effects, but all first-order white-box attacks agree on their Macro effects.

TABLE III: Adversarial attack's Macro effect on fully neural network (NN) and exponential hazard (EXP) models w.r.t. Hawkes process parameters, μ, α, and β.

	Fully neural network			**Exponential hazard**		
	$\boldsymbol{\mu}$ (e-02)	$\boldsymbol{\alpha}$ (e-01)	$\boldsymbol{\beta}$ (e-01)	$\boldsymbol{\mu}$ (e-02)	$\boldsymbol{\alpha}$ (e-01)	$\boldsymbol{\beta}$ (e-01)
Test set	1.29	8.74	9.11	1.29	8.74	9.11
First-order attacks	99.8	10.0	13.0	99.8	10.0	13.0
GD	99.6	9.9	9.08	99.6	9.9	10.2
TM	99.6	9.99	11.5	99.6	9.99	11.0
Prediction in presence of adversarial attacks						
No Attack	1.25	8.79	8.43	1.32	8.70	9.43
PGD	1.20	8.82	7.94	1.32	8.71	9.19
iFGSM	1.17	8.84	7.77	1.32	8.71	9.16
$R+FGSM$	1.18	8.83	7.84	1.32	8.72	9.11
miFGSM	1.18	8.84	7.80	1.32	8.71	9.15
FGSM	1.18	8.83	7.83	1.32	8.71	9.24
SM	1.23	8.81	8.19	1.32	8.70	9.35
GD	1.12	8.88	7.372	1.23	8.81	8.20
TM	2.73	7.30	53.5	1.54	8.41	10.00

In Fig. 3 and Fig. 4 we represent the effect of adversarial attacks on Hawkes parameters. As presented, all first-order white-box attacks, PGD and surrounded gray area, both against the NN model and the EXP model, share the underlying parameters, μ, α, and β and agree on their Macro effects. Additionally, although all adversarial events have a greater intensity range in comparison to real events, but both neural models are successful in maintaining the range of predicted intensity, even in presence of adversarial attacks. However, if the adversarial sets, e.g., PGD, be fed to a parametric model, the parametric model will failed in modeling them.

Considering single-point attacks, e.g., GD, they are not following the rest of adversarial attacks, but are more proximate to each other. The algorithm responsible for generating GD samples can explain the shift in parameters and intensity functions of TM and GD since the GD attack can be considered as a steady version of TM. To evaluate the transferability of white-box attacks against NN and EXP models, for each attack, we perform attacks against independently trained larger networks, more complexity, and then deploy the adversarial set to our original targeted models. To increase the complexity, we try hyper-parameters as discussed in the transferability section III-C. For the exponential hazard model, the "RNN units" is the only available hyper-parameter. In Fig. 5, transferability power of attacks against our models is shown. Transferability investigation implies the susceptibility of both models to iFGSM adversarial attack, the strongest attack w.r.t. predictive performance, is not increased in transferring. For the EXP model, to have an single-step effective attack, the adversarial samples require to be crafted in a white-box setting. Still, for the NN model, the adversarial goal is feasible in both black-box (transferred) and white-box settings.

Finally, in Table IV, models' performance regarding non-stationary abrupt changes, Covid-19 pandemic's impact on reported crimes to Chicago Police Department in 2020, is reported. According to the results, fully neural network model performs better in modeling crimes, but the exponential hazard model is more robust (% of change in MNLL) against non-stationary abrupt changes, although it's still sensitive.

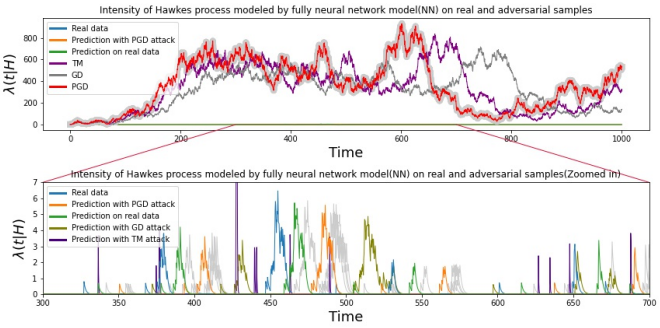

Fig. 3: Adversarial attack's Macro effect on the fully neural model (NN) parametric modeling performance. Top: Conditional intensity of real events (Blue), predicted events (Green), and adversarial events generated by PGD attack (Red), GD (Gray), and TM (Purple), respectively. Bottom: Enlarged view. Note that adversarial events' intensity, $\lambda(t|H)$, range is far from the real and predicted events. Here, $\lambda(t|H)$ of all other first-order attacks and corresponded predictions have been presented in "light gray" as they are tight to PGD. Additionally, we present $\lambda(t|H)$ of GD and predicted event arrivals in the presence of GD in "Olive". The same for TM attack is shown in "Indigo".

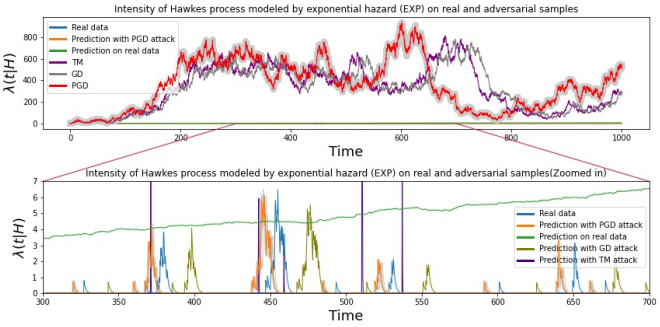

Fig. 4: Adversarial attack's Macro effect on the exponential hazard (EXP) parametric modeling performance.

VI. Conclusion

We have proposed and studied several white-box and black-box adversarial attacks on two state-of-the-art deep learning point processes that provide non-parametric modeling of temporal point processes, and investigate the transferability of proposed adversarial attacks. Adversarial samples creation is a critical step, especially when the prediction from such models is utilized in safety and cost-critical applications. Moreover, we have studied the performance of the models facing non-stationary abrupt changes such as Covid-19. Finally, we have examined how Hawkes process's parameters, are vulnerable to adversarial attacks. According to experimental results, one can attack both predictive and parameter-estimation performance of neural point processes with a small perturbation. Addition-

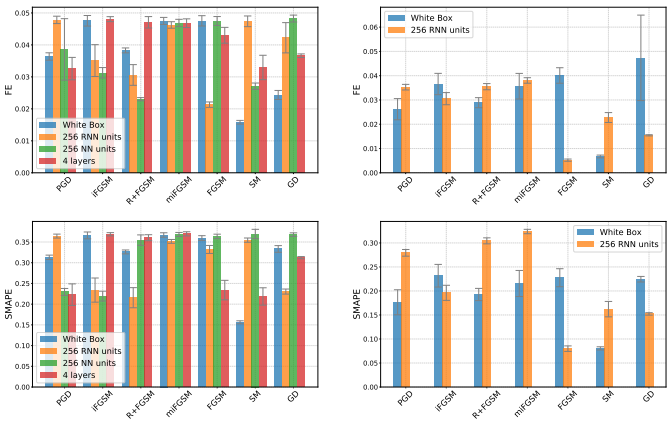

Fig. 5: Fooling error (Top) and Symmetric Mean Accuracy Percentage Error (Bottom) of each attack in the white box and semi-black-box settings against (left) fully neural model (NN) and (right) exponential kernel hazard model (EXP).

TABLE IV: The performance of the models on crime prediction in 2019 and 2020. fully neural network hazard function (Left half), and exponential hazard model (Right half)

	Fully neural network		**Exponential hazard**	
Data	MNLL (e+00)	MAE (e+01)	MNLL (e+00)	MAE (e+01)
Crimes (2019)	5.08	9.66	5.672	11.06
Crime (2020)	5.44	13.1	5.929	14.62

ally, results imply the models are less sensitive to transferable attacks, and are not robust to non-stationary changes in the data such as Covid-19. Exploring defense mechanisms to improve the models' robustness is the direction of our future work.

VII. Acknowledgements

This research was supported by AFOSR MURI grant FA9550-22-1-0380 and NSF grant ATD-2124313. Bao Wang is sponsored by NSF grants DMS-1924935, DMS-1952339, DMS-2110145, DMS-2152762, and DMS-2208361, and DOE grant DE-SC0021142.

References

[1] S. Zhu, H. S. Yuchi, and Y. Xie, "Adversarial anomaly detection for marked spatio-temporal streaming data," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8921–8925.

[2] R. Shah, K. Muralidharan, and A. Parajuli, "Temporal point process models for nepal earthquake aftershocks," *International Journal of Statistics and Reliability Engineering*, vol. 7, no. 2, pp. 275–285, 2020.

[3] T. Murayama, S. Wakamiya, and E. Aramaki, "Fake news detection using temporal features extracted via point process," *arXiv preprint arXiv:2007.14013*, 2020.

[4] F. Adesola, A. Azeta, A. A. Oni, and F. Chidozie, "Forecasting violent crime hotspots using a theory-driven algorithm," *International Journal of Engineering Research and Technology (IJERT)*, 2019.

[5] S. Xiao, J. Yan, S. M. Chu, X. Yang, and H. Zha, "Modeling the intensity function of point process via recurrent neural networks," 2017.

[6] H. Mei and J. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," *arXiv preprint arXiv:1612.09328*, 2016.

[7] T. Omi, N. Ueda, and K. Aihara, "Fully neural network based model for general temporal point processes," *CoRR*, vol. abs/1905.09690, 2019.

[8] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha, "Wasserstein learning of deep generative point process models," *arXiv preprint arXiv:1705.08051*, 2017.

[9] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1555–1564.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] E. Lewis and G. Mohler, "A nonparametric em algorithm for multiscale hawkes processes," *Journal of Nonparametric Statistics*, vol. 1, no. 1, pp. 1–20, 2011.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[13] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "The robustness of deep networks: A geometrical perspective," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 50–62, 2017.

[14] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, p. 828–841, Oct 2019.

[15] K. Alparslan, Y. Alparslan, and M. Burlick, "Adversarial attacks against neural networks in audio domain: Exploiting principal components," 2021.

[16] S. Khorshidi, M. Al Hasan, G. Mohler, and M. B. Short, "The role of graphlets in viral processes on networks," *Journal of Nonlinear Science*, pp. 1–16, 2018.

[17] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, and H. Zha, "Fake news mitigation via point process based intervention," 2017.

[18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Adversarial attacks on deep neural networks for time series classification," *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul 2019.

[19] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[22] G. R. Mode and K. A. Hoque, "Adversarial examples in deep learning for multivariate time series regression," 2020.

[23] F. P. Schoenberg, "Introduction to point processes," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[24] Y. Lee, P. J. Laub, T. Taimre, H. Zhao, and J. Zhuang, "Exact simulation of extrinsic stress-release processes," 2021.

[25] S. Khorshidi, J. G. Carter, and G. Mohler, "Repurposing recidivism models for forecasting police officer use of force," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020.

[26] M.-A. Rizoiu, Y. Lee, S. Mishra, and L. Xie, "A tutorial on hawkes processes for events in social media," 2017.

[27] T. Omi, N. Ueda, and K. Aihara, "Fully neural network based model for general temporal point processes," *CoRR*, vol. abs/1905.09690, 2019.

[28] A. T. Nguyen and E. Raff, "Adversarial attacks, regression, and numerical stability regularization," 2018.

[29] K. Gupta, J.-C. Pesquet, B. Pesquet-Popescu, F. Kaakai, and F. D. Malliaros, "An adversarial attacker for neural networks in regression problems," in *AISafety@IJCAI*, 2021.

[30] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," 2017.

[31] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: attacks & defenses," 2017.

[32] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Transactions on Image Processing*, 2010.

[33] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," 2015.

[34] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013.

[35] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," 2017.