LAVA: Longitudinal Adversarial Attack on Electronic Health Records Data

Sungtae An
Georgia Institute of Technology
Atlanta, Georgia
stan84@gatech.edu

Cao Xiao
IQVIA
Cambridge, Massachusetts
cao.xiao@iqvia.com

Walter F. Stewart[*]
HINT Consultants
Walnut Creek, California
wfs502000@yahoo.com

Jimeng Sun
Georgia Institute of Technology
Atlanta, Georgia
jsun@cc.gatech.edu

ABSTRACT

Although deep learning models trained on electronic health records (EHR) data have shown state-of-the-art performance in many predictive clinical tasks, the discovery of adversarial examples (i.e., input data that are engineered to cause misclassification) has exposed vulnerabilities with lab and imaging data. We specifically consider adversarial examples with longitudinal EHR data, an area that has not been previously examined because of the challenges with temporal high-dimensional and sparse features. We propose Longitudinal AdVersarial Attack (LAVA), a saliency score based adversarial example using a method that requires a minimal number of perturbations and that automatically minimizes the likelihood of detection. Features are selected and modified by jointly modeling a saliency map and attention mechanism. Experimental results with longitudinal EHR data show that LAVA can substantially reduce model performance for attention-based target models (from AUPR = 0.5 to AUPR = 0.08).

CCS CONCEPTS

• Computing methodologies → Supervised learning by classification; **Neural networks**; **• Applied computing → Health informatics**.

KEYWORDS

Health analytics; predictive model; adversarial examples; attention mechanism; neural networks

ACM Reference Format:
Sungtae An, Cao Xiao, Walter F. Stewart, and Jimeng Sun. 2019. LAVA: Longitudinal Adversarial Attack on Electronic Health Records Data. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313528

[*] Work done at Sutter Health.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313528

1 INTRODUCTION

Electronic health record (EHR) data from millions of patients are now routinely collected across diverse healthcare systems and physician practices, that can be represented as a longitudinal patient-level history of diagnoses, laboratory test results, and medication prescriptions orders, procedure orders, among other events and accompanied by demographic and other features. Deep learning models trained on these data are yielding high levels of performance for many clinical tasks such as diagnostic classification [15], disease detection [5], risk prediction [24]. In some instances, the performance of these models exceed the capabilities of experienced physicians in head-to-head comparisons [2]. However, the discovery of "adversarial examples" has exposed vulnerabilities in deep learning models across various application domains [9]. The connected EHR systems are under risk of adversarial attacks for a number of reasons including manipulations of pharmaceutical and device approvals, medical reimbursement decisions, etc [7]. As a major means of adversarial attacks, the "adversarial examples" are crafted to cause diagnostic errors via perturbation of EHR data, and further trigger significant consequences in treatment or other patient outcomes. Therefore, the risks from adversarial attacks need to be mitigated. Ultimately, a good understanding of such adversarial examples on longitudinal EHR can help patients and EHR administrators to proactively prevent the high-risk attacks to their records.

Adversarial examples have received much attention in computer vision [13, 19, 21] and natural language processing [12, 29], but relatively little work has been done with a focus on EHR data. Finlayson et al. [7] demonstrated vulnerability of convolutional neural network models trained with medical imaging datasets, replicating prior work on imaging from other industries. Sun et al. [26] proposed a recurrent neural network (RNN) based time-preferential minimum attack model to identify susceptible locations in clinical time series data focused on real-valued continuous variables such as vital sign and lab measurements. Despite all these efforts, adversarial attacks on longitudinal EHR data on neural network models is still fairly open. More generally, challenges to devising adversarial examples for longitudinal EHR data include:

- *High-dimensional, discrete and sparse feature space*: Unlike most other domains such as image and audio waveform, EHR data are characterized by high-dimensional, discrete, and extremely sparse feature space. As many as 80% to 90% of the 10,000 or more features (i.e., diagnosis, medication, and procedure codes) have non-null values less than 1% of the time. Thus, existing crafting algorithms for continuous or dense data cannot be directly applied because they assume perturbations were broadly spread over the entire feature space.

- *Feature significance and relevance*: Each medical code differs in clinical significance and relevance to prediction tasks. For example, diagnostic codes for fever and heart failure inform the different level of risks in terms of morbidity and mortality risks. Perturbation of critical features such as heart failure could be readily detected even though they are the most effective attacks.
- *Temporal significance*: Patients' health conditions evolve over time. The temporality encoded in time-stamped medical events reveals important information on impending patient health conditions, with differential significance on prediction outcomes. For example, more recent events are likely to have a stronger influence on the prediction result. Thus adversarial attacks on more recent events are more likely to be detected.

To address these challenges, we propose LAVA (**L**ongitudinal **A**d**V**ersarial **A**ttack), a saliency score based method for discrete and sequential EHR data. LAVA introduces a minimal number of perturbations while also avoiding the selection of features and clinical encounters whose modifications are highly likely to be detected. This is achieved by jointly modeling the saliency map and attention mechanism. We specifically propose LAVA with the following technical contributions.

- *Efficient crafting with a bidirectional saliency map*: We proposed a new bidirectional saliency map that searches candidate features from both (+/−) altering directions simultaneously.
- *Low detectability with an attention mechanism*: LAVA automatically avoids choosing features that have high relevance to the prediction targets. This is achieved via a dual-level attention mechanism that can highlight and avoid important visits and important features within those visits.

In experimental modeling with real-world data show that LAVA substantially reduces performance (i.e., area under the precision-recall curve or AUPR) for the white-box[1] type of attention-based target models from 0.5 to 0.08, and achieves comparable performance reductions for other gray-box[2] target models, while providing minimal to non-detectable perturbations.

2 RELATED WORK

Adversarial examples can be crafted in different ways and categorized by the norm they use. In general, L_p-norm, $\|\cdot\|_p$ measures the distance between an original input $\mathbf{x}$ and the corresponding adversarial example $\tilde{\mathbf{x}}$. The choice of L_p (e.g., L_0, L_1, L_2, or L_∞) is related to the problem domain. Goodfellow et al. [9] proposed the Fast Gradient Sign Method (FGSM), a widely used method that generates untargeted adversarial examples optimized under the L_∞ norm. FGSM works by adding perturbations of size ϵ to the direction of increasing loss function $J(\theta, \mathbf{x}, y)$ used to train the model F with the model parameters θ, the input $\mathbf{x}$, and the target y. Kurakin et al. [14] proposed an iterative variant of FGSM called Projected Gradient Descent (PGD), which shows better generalization and performance for most cases. PGD is a universal adversary among first-order approaches [17]. Moosavi Dezfooli et al. [20] proposed an approach that uses a directional derivative to find a distance from the input data point in the feature space to the decision boundary, crafting adversarial examples that minimize perturbations.Papernot et al. [22] proposed Jacobian-based Saliency Map Approach (JSMA) that finds a feature that has the highest overall impact on leading a classifier to an aimed wrong class label at each iteration under the L_0 (or L_1 according to the termination condition) norm constraint.

L_∞ norm constrained approaches (e.g., FGSM and PGD) add perturbations to each dimension of the entire feature space; thus, it is difficult to apply these methods directly to EHR data given the sparseness of the feature space, where non-zero values are rare. We show the perturbation distribution generated by PGD from EHR data in Section 5. For both the discrete and sparse feature space we selected L_0 or L_1 as the options for the target domain, where the proposed method LAVA is optimized under L1 objective.

3 BACKGROUND

3.1 Jacobian-based Saliency Map

The Jacobian-based Saliency Map Approach (JSMA) [22] finds a feature that has the highest overall impact on leading a classifier to a targeted wrong class label at each iteration. Given $F : \mathbb{R}^d \mapsto \mathbb{R}^s$, a neural network that maps an input vector $\mathbf{x} \in \mathbb{R}^d$ to probabilities over s classes, x_k is the k-th element of the input vector $\mathbf{x}$, and F_r is the model output probability for the class r w.r.t. the input $\mathbf{x}$, the Jacobian matrix of the model is computed as the forward derivative of the neural network, where the element at the k-th column and the r-th row is denoted as $\mathbf{J}_{rk}$, is given by $\mathbf{J}_{rk} = \frac{\partial F_r}{\partial x_k}$. Assuming that t is the target class that an adversary wants the neural network model incorrectly classify the input $\mathbf{x}$ into, a *target class saliency* A_k and an *overall non-target classes saliency* B_k for feature k are given by

$$A_k = \mathbf{J}_{tk} = \frac{\partial F_t}{\partial x_k}, \quad B_k = \sum\nolimits_{r \neq t} \mathbf{J}_{rk} = \sum\nolimits_{r \neq t} \frac{\partial F_r}{\partial x_k}. \tag{1}$$

We can then construct a saliency map $S(\mathbf{x}, t)$ based on A_k and B_k to search candidate features. The JSMA is applicable to craft adversarial example from EHR data since it perturbs one feature at each iteration. However, it only searches along one direction (i.e., either increasing or decreasing) and thus is not adequately efficient given the very high feature space in EHR data.

3.2 Neural Attention Mechanism

The attention mechanism is increasingly used to understand what part of spatial or temporal information contributes most in predicting target values. It is widely used applications for computer vision [18, 28], natural language processing [1, 11], and speech recognition [6]. When introduced to EHR modeling, attention weights indicate the degree to which past clinical events predict disease onsets or future events [3, 16]. Choi et al. [4] proposed a model architecture with a dual-level attention mechanism, RETAIN, which calculates attention weights at the visit and feature (embedding) levels to compose the context vector for the final predictions. In this current

[1] White-box means both training data and the model architecture are available to the attackers.

[2] Gray-box refers to the setting where model architectures are unknown but training data are available to the attackers.

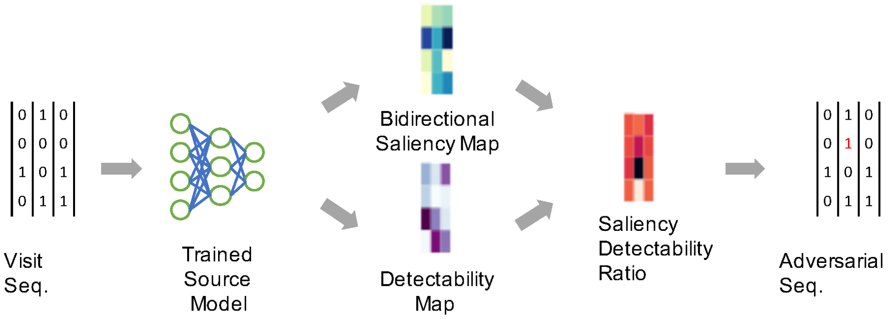

Figure 1: Crafting iteration of LAVA. LAVA balances the effectiveness of the attack through the bidirectional saliency and the detectability of the attack over longitudinal EHR data. Given a trained model and an input visit sequence of a patient, LAVA calculate both the bidirectional saliency score and the detectability cost for each feature at each visit. LAVA increases or decreases the binary feature value that has the highest saliency-to-detectability ratio at a specific visit. This procedure is iteratively performed until it satisfies stopping criteria.

work, we have adopted the RETAIN architecture to capture both visit and feature level significance and use this information to estimate the likelihood of perturbations to be discovered by a target model or defender.

4 LAVA: LONGITUDINAL ADVERSARIAL ATTACK

Given longitudinal EHR data from N patients, denote $\mathbf{x}^{(n)}$ as the clinical trajectory of patient n, which is characterized by a sequence of T^n hospital visits. Then $\mathbf{x}^{(n)}$ can be formulated as given by Eq. 2.

$$\mathbf{x}^{(n)} = \left\{\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \dots \mathbf{x}_{T^n}^{(n)}\right\} \tag{2}$$

where $\mathbf{x}_i^{(n)} \in \mathbb{R}^d$ denotes the concatenation of medical codes (often binary encoded) of the i-th visit made by patient n. In the following, we omit the superscript (n) to reduce clutter (e.g., represent $\mathbf{x}^{(n)}$ as $\mathbf{x}$, and $\mathbf{x}_i^{(n)}$ as $\mathbf{x}_i$). We also denote $x_{i,k}$ as the k-th dimension of the i-th visit $\mathbf{x}_i$ for a patient. Then, for a given patient records $\mathbf{x}$, a trained model F, and an original label $y = F(\mathbf{x})$, crafting adversarial example from longitudinal EHR requires finding a perturbation $\delta_\mathbf{x}$:

$$\underset{\delta_\mathbf{x}}{\operatorname{argmin}} \|\delta_\mathbf{x}\|_p \quad \text{s.t. } F(\mathbf{x} + \delta_\mathbf{x}) = \tilde{y} \neq y \tag{3}$$

The resulting adversarial example $\tilde{\mathbf{x}} = \mathbf{x} + \delta_\mathbf{x} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots \tilde{\mathbf{x}}_T\}$ can have $\tilde{\mathbf{x}}_i \neq \mathbf{x}_i$ and $\tilde{x}_{i,k} \neq x_{i,k}$ for any k-th feature in any i-th visit.

4.1 Overview

We propose LAVA as an adversarial examples crafting method for discrete and sequential EHR data. LAVA iteratively searches features from selected clinical visits that are suitable as adversarial examples as defined by a higher ratio of the sensitivity to fool a given model over the likelihood of the attack being detected. LAVA executes the following sequential procedure.

- Computes the **bidirectional saliency score** of each feature at each visit for perturbation to measure suitability as an adversarial example.
- Computes the **detectability cost**, which is the likelihood that modification to a feature will be detected.
- Calculates the **saliency-to-detectability ratio** to determine which feature at a specific visit is optimal to be perturbed.

This procedure is iteratively performed until it satisfies stopping criteria. Figure 1 depicts the overall process pipeline for an iteration of LAVA, and the details of each step are described in the following sections.

4.2 Bidirectional Saliency Score

To find the most prominent feature that contributes to the current classification result, we propose the *bidirectional saliency score* that rearranges and combines the computation of increasing and decreasing saliency map [22] with a minimal additional computational cost. We first derive a bidirectional saliency map for non-sequential data first, then extend it to sequential data.

Non-sequential Data. Given (1), the bidirectional saliency map $S(\mathbf{x}, t)$ for the input $\mathbf{x}$ and the target label t is given by (4).

$$S(\mathbf{x}, t)[k] = \operatorname{sgn}(B_k) \min(0, A_k B_k) \tag{4}$$

Here a positive valued $A_k B_k$ indicates two possible cases: both A_k and B_k are either positive or negative simultaneously, such that feature x_k has the impact on the probabilities of target class and non-target classes toward the same direction (e.g., increase or decrease). To avoid this ambiguity, we limit $A_k B_k$ to negative values only where the feature x_k has opposing directional effect on the target class and on the non-target classes. When this condition arises, a saliency score has a sign that is opposite to the sign of B_k but same as the sign of A_k since the resulting value of $\min(0, A_k B_k)$ is always negative. Consequently, this single bidirectional saliency map $S[k]$, omitting $\mathbf{x}$ and t, indicates which direction of perturbation on the k-th dimension of the input feature leads to a higher probability that the input is classified as the target class t with its relative strength given by the value of $S[k]$.

Sequential Data. For sequential data such as the longitudinal EHR sequence in our study, we firstly represent it as Eq. 2, then we adopt a RETAIN architecture [4], an RNN-based attention model, to be the source model for crafting adversarial examples, and extend the bidirectional saliency method to handle sequential data. In particular, we extend the forward derivative, the Jacobian matrix whose element is given by $\mathbf{J}_{rk}$ in Section 3, to a sequential input for a recurrent neural network. Computing the forward derivative through a recurrent neural network is straightforward; any recurrent connections can be unrolled and transformed into a directed acyclic graph (DAG), and the partial derivatives of output with respect to input at each visit on the unrolled computational graph can easily be taken [8]. Thus, a Jacobian matrix can be computed for each visit, and they can be represented together as a tensor form with an additional time axis. Consequently, each element of the forward derivative and the corresponding target and non-target saliency are formulated as (5)–(7).

$$\mathbf{J}_{rik} = \frac{\partial F_r}{\partial x_{i,k}} \tag{5}$$

$$A_{ik} = \mathbf{J}_{tik} = \frac{\partial F_t}{\partial x_{i,k}} \tag{6}$$

$$B_{ik} = \sum\nolimits_{r \neq t} \mathbf{J}_{rik} = \sum\nolimits_{r \neq t} \frac{\partial F_r}{\partial x_{i,k}} \tag{7}$$

where $x_{i,k}$ is the k-th coordinate of the input vector at time i. As a result, a bidirectional saliency score map where each element is a score for each k-th feature at each i-th input $\mathbf{x}_i$ of a sequential input $\mathbf{x}$ can be obtained simply by (8).

$$S[i,k] = \text{sgn}(B_{ik}) \min(0, A_{ik}B_{ik}) \tag{8}$$

Assuming that each visit vector $\mathbf{x}_i$ is a binary vector where each $x_{i,k}$ can have only either 1 or 0, we apply a restriction that a feature valued at 0 will not decrease and valued at 1 will not increase. Thus the bidirectional saliency score for sequential data is given by (9).

$$S[i,k] = \begin{cases} \max(0, \text{sgn}(B_{ik}) \min(0, A_{ik}B_{ik})), & \text{if } x_{i,k} = 0 \\ \min(0, \text{sgn}(B_{ik}) \min(0, A_{ik}B_{ik})), & \text{if } x_{i,k} = 1 \end{cases}. \tag{9}$$

4.3 Detectability Cost

We formulate the *detectability cost* using the attention mechanism to measure the imperceptibility of perturbations where the lower cost indicates the lower chance that a perturbation will be detected. The attention mechanism is a widely used technique to pay more 'attention' to features influential on model decisions; significant changes in the amount of 'attention' on each event reflect either a critical event has been newly added or an existing important event has been removed. In particular, the impact of each $x_{i,k}$ on the final classification result is measured from the two-level attention weights by (10) following the derivation from [4]:

$$w_{ik} = \alpha_i \mathbf{W}(\boldsymbol{\beta}_i \odot \mathbf{W}_{\text{emb}}[:,k]) \tag{10}$$

where α_i is the attention weight on the i-th visit, $\boldsymbol{\beta}_i$ is an attention weight vector for all features $x_{i,k}$ of the i-th visit, $\mathbf{W}$ is the output weight matrix, $\mathbf{W}_{\text{emb}}$ is the weight matrix at the embedding layer, and $\odot$ denotes element-wise multiplication. Here we assume that an input vector $\mathbf{x}_i$ at time i is a binary vector. Similar to the computation of the saliency score in the previous section, the forward derivative of each w_{jl} with respect to each input feature $x_{i,k}$ can be obtained from the unrolled computational graph. Then, we define the detectability cost by

$$D[i,k] = \sum_{j,l} \frac{\partial w_{jl}}{\partial x_{i,k}} \tag{11}$$

which captures the overall amount of change of the impact by all features and all visits when the feature $x_{i,k}$ is perturbed.

4.4 Saliency-to-Detectability Ratio

Last, we calculate *Saliency-to-Detectability Ratio* (*SDR*) at each iteration, which is defined as a ratio between the normalized saliency score and the normalized detectability cost as given by (12)–(13).

$$SDR[i,k] = \frac{\hat{S}_{ik}}{\hat{D}_{ik}} \tag{12}$$

where

$$\hat{S}_{ik} = \frac{S_{ik}}{\max_{j,l} \left|S_{jl}\right|}, \quad \hat{D}_{ik} = \frac{D_{ik}}{\max_{j,l} \left|D_{jl}\right|}. \tag{13}$$

Consequently, the feature $x_{i,k}$ that has the maximum value of $SDR[i,k]$ is perturbed in the direction of the sign of $SDR[i,k]$ at each iteration. To conclude, we summarize the pseudo code of LAVA in Algorithm 1. The source code of LAVA is publicly available at https://github.com/ast0414/lava

Algorithm 1 LAVA (Longitudinal AdVersarial Attack)

F is a trained model, $\mathbf{x}$ is an input sequence, $\tilde{\mathbf{x}}$ is a crafted adversarial sequence, t is a target class/label, and M is the maximum number of distortions

```
procedure LAVA($F, \mathbf{x}, t, M$)
    $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$
    count ← 0
    $y, w \leftarrow F(\tilde{\mathbf{x}})$ ▷ $y$ is a model output and $w$ is a contribution matrix
    while count < $M$ and $y \neq t$ do
        Initialize $S$ and $D$ ▷ a bidirectional saliency map $S$ and a detectability map $D$
        for all $i$-th visit and $k$-th feature do
            $A_{ik} \leftarrow \frac{\partial F_t}{\partial \tilde{x}_{i,k}}$
            $B_{ik} \leftarrow \sum_{r \neq t} \frac{\partial F_r}{\partial \tilde{x}_{i,k}}$
            $S[i,k] \leftarrow \text{sgn}(B_{ik}) \min(0, A_{ik}B_{ik})$
            if $x_{i,k} == 0$ then
                $S[i,k] \leftarrow \max(0, S[i,k])$
            else
                $S[i,k] \leftarrow \min(0, S[i,k])$
            $D[i,k] \leftarrow \sum_{j,l} \frac{\partial w_{jl}}{\partial x_{i,k}}$
        $\hat{S}_{ik} \leftarrow S_{ik} / \max_{j,l} \left|S_{jl}\right|$
        $\hat{D}_{ik} \leftarrow D_{ik} / \max_{j,l} \left|D_{jl}\right|$
        for all $i$-th visit and $k$-th feature do
            $SDR[i,k] \leftarrow \hat{S}_{ik} / \hat{D}_{ik}$
        $i^*, k^* \leftarrow \text{argmax}_{i,k} SDR[i,k]$
        $\tilde{x}_{i^*,k^*} \leftarrow \tilde{x}_{i^*,k^*} + 1 \cdot \text{sgn}(SDR[i^*,k^*])$
        $y, w \leftarrow F(\tilde{\mathbf{x}})$
        count ← count + 1
    return $\tilde{\mathbf{x}}$
```

Table 1: Descriptive statistics of EHR dataset

# total patients	30,742	Avg. # visits per patient	18.68
# case patients	3,409	Max. # visits per patient	246
# total visits	574,108	Avg. # codes per visit	3.27
# unique codes	17,081	Max. # codes per visit	47

Table 2: Model performances on the clean test set

Model	AUROC	AUPR	F1 Score	Accuracy
RETAIN-WHITE	0.851	0.495	0.465	0.821
RETAIN-GRAY	0.849	0.493	0.461	0.817
RNN	0.837	0.456	0.444	0.809
MLP	0.821	0.388	0.426	0.797

5 EXPERIMENTS

In this section, we evaluated the sequences with adversarial examples using different types of neural network models.

5.1 Setup

Data. We conducted all experiments using EHR data provided by Sutter Health. The dataset was originally constructed for a heart failure (HF) onset prediction study that consists of 30K patients (age 50-89) including 3K HF case patients chosen by a set of criteria described by Gurwitz et al. [10], Vijayakrishnan et al. [27] and 27K control patients matched by a set of criteria described by Choi et al. [5]. Patients data over an 18 month period were used in this study. The summary statistics of the data is described in Table 1.

Source model. To craft adversarial examples, we used the RETAIN [4] architecture with two independent GRU layers, each of which includes 128 hidden units and embedding layer of 128 dimensions.

Target models. To evaluate multiple aspects of the crafted adversarial examples, we considered the following white-box and gray-box target models. The white-box target model is identical to the source model that crafts adversarial examples. The gray-box models are ones with different architectures and/or different hyperparameters while they are trained using the same train data as the source model.

- *RETAIN-WHITE*: Pure white-box attack was evaluated on the exact same architecture and trained parameters as the source model.
- *RETAIN-GRAY*: A RETAIN model with a different set of hyperparameters was trained and evaluated on attacks. It has 256 hidden units for each GRU layer for visit and feature attention, and an embedding layer of 256 dimensions was used.
- *RNN*: An RNN model without attention mechanism. It includes two GRU layers, each consisting of 128 hidden units, and an embedding layer of 128 dimensions.
- *MLP*: A multi-layer perceptron (MLP) model that was trained on aggregated feature vectors where a patient is represented by a single vector without temporal information. Then, the crafted sequential adversarial examples were also aggregated for each patient and passed into the trained MLP. The MLP model consists of three hidden layers, each of which consists of 128 hidden units.

Baseline methods. We compare LAVA with the following crafting algorithms.

- LAVA *w/o detectability*: We used non-penalized saliency scores only to find a candidate feature at each iteration. In other words, this algorithm is equivalent to LAVA where the detectability costs for all feature over the visits are equal to one.
- *PGD*: Adversarial examples were crafted by Projected Gradient Descent [13], where we set the step size as 1 and limit feature space as binary.
- *Random*: At each iteration, a visit was randomly selected from the sequence of visits, and a feature, a medical code, was chosen also randomly to be perturbed.

Implementation details. All models were implemented using PyTorch 0.3 [23] and trained with a system equipped with Intel Xeon E5-2683v4 CPU, 756GB RAM, and Nvidia Tesla P100 GPU. We used Stochastic Gradient Descent (SGD) with momentum [25] for optimization. The momentum value was set to 0.9. The initial learning rate was set to 0.01 and was reduced by a factor of 10 once the learning stagnated.

5.2 Attack Evaluation

As a contrast, we first show how these target models perform on unperturbed data. Table 2 shows that all models achieved accurate predictions of a future HF diagnosis when using unperturbed test data. Figure 2 compares the performance of different craft algorithms by the extent to which they reduce prediction performance as defined by the Area Under Precision-Recall curve (AUPR).

- For attention based target models such as RETAIN-WHITE and RETAIN-GRAY, LAVA achieved the best performance drop and was able to beat the state-of-the-art gradient-based method, PGD, after 12 iterations.
- For some models (e.g., RNN and MLP) and during early iterations of attention models, LAVA caused less AUPR drop than LAVA w/o detectability, the non-penalized saliency score method. This is due to the fact that saliency score methods often perturbed the most critical feature, while LAVA avoids such perturbation. During interactive attack and defense, perturbing most critical features will again be easily caught. While LAVA address this concern, it still achieved good AUPR drops.
- Although PGD showed good performances in some settings, it perturbs a large number of features (1,650 per visit in our experiment) even in the first iteration. While LAVA is cost-sensitive, it only perturbs one feature at a time.
- Moreover, the random perturbations barely have any influence on the performance for all settings.

5.3 Perturbation Analysis

We evaluated the perturbations made by each algorithm to determine effectiveness and the ease of detection (i.e., from the EHR system perspective that is a target of an adversarial attack). Figure 3 shows how the perturbations in the adversarial examples crafted by each algorithm are distributed over the visit and feature

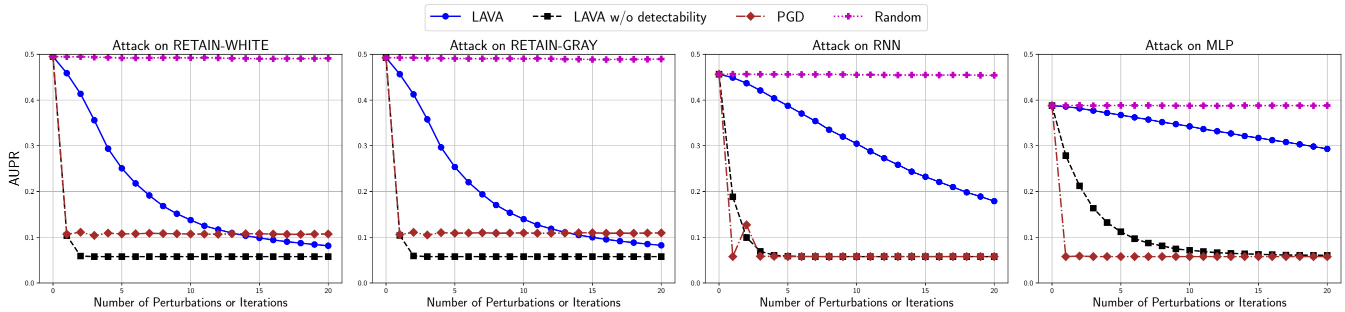

Figure 2: Comparison of the different types of attack on each target model.

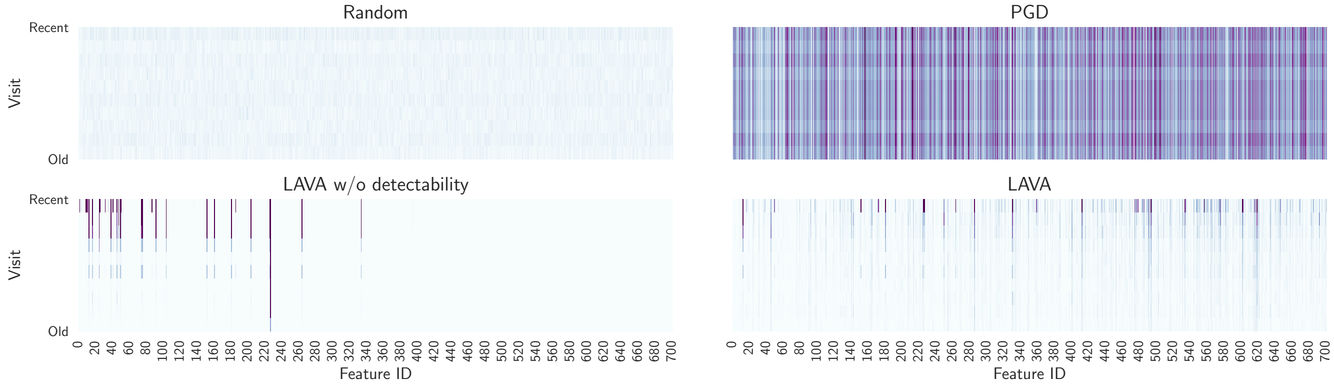

Figure 3: Comparison of perturbations by the different types of attack. The x-axis represents features ordered by feature ID. The y-axis represents visits in chronological order from the bottom to the top. Each heatmap depicts how the adversarial perturbations by each algorithm are distributed over the visits and features in the sequential EHR data. The white color means no perturbation and the darker color represents the more perturbations.

dimension of the test sequential EHR data. Random attacks spread perturbations over the entire visit and feature space, but each perturbation holds no meaningful attraction for the models since it was crafted in a purely random manner. PGD attacks also crafted perturbations over the entire space and are densely distributed, influencing almost every feature in every visit. PGD attacks would be easily discovered by a defender (e.g., an administrator of the EHR system that might be human or machine). Moreover, PGD attacks may be treated as a system malfunction since it may seem the data themselves are scrambled. On the contrary, the perturbations in the adversarial examples crafted using non-penalized saliency scores are concentrated on a few certain features in relatively recent visits. Thus, the administrator of a machine learning based EHR system can monitor and pay attention to only those particular features to make provisions against potential adversarial attacks by non-penalized saliency methods. Adversarial examples crafted by LAVA have perturbations spread wider than those made by the algorithm without detectability costs although the recent visits are still highlighted slightly more than the others. Hence, the administrator may need to monitor the system more carefully to figure out the existence of adversarial attacks; otherwise, it might be misunderstood for random noises that are naturally inherent in the data. In conclusion, it is shown that LAVA crafted the adversarial examples for sequential EHR data, where it is difficult to detect their existence in terms of the distribution of the perturbations over both visit and feature spaces.

6 CONCLUSION

In this work, we propose LAVA to craft adversarial examples on longitudinal EHR data. We compared the attack result by LAVA with other attacks. It is shown that the adversarial examples crafted by LAVA perform effective attacks toward white-box and gray-box target models. Moreover, it is also shown that the adversarial examples crafted by LAVA are relatively well spread over the visits and the feature space; it would be more difficult to detect such attacks generated by LAVA than other attacks. As future work, we plan to design and compare different defense mechanisms to provide better guidance on how to alleviate the adversarial attacks on longitudinal EHR data.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation, award IIS-1418511 IIS-1838042 and CCF-1533768, the National Institute of Health award 1R01MD011682-01 and R56HL138415.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.

[3] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: Graph-based attention model for healthcare representation learning. In *SIGKDD*.

[4] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3504–3512.

[5] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association* 24, 2 (2016), 361–370.

[6] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*. 577–585.

[7] Samuel G Finlayson, Isaac S Kohane, and Andrew L Beam. 2018. Adversarial Attacks Against Medical Deep Learning Systems. *arXiv preprint arXiv:1804.05296* (2018).

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[10] Jerry H Gurwitz, David J Magid, David H Smith, Robert J Goldberg, David D McManus, Larry A Allen, Jane S Saczynski, Micah L Thorp, Grace Hsu, Sue Hee Sung, et al. 2013. Contemporary prevalence and correlates of incident heart failure with preserved ejection fraction. *The American journal of medicine* 126, 5 (2013), 393–400.

[11] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. 1693–1701.

[12] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[14] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).

[15] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. 2016. Learning to diagnose with LSTM recurrent neural networks. In *ICLR*.

[16] Tengfei Ma, Cao Xiao, and Fei Wang. 2018. Health-ATM: A Deep Architecture for Multifaceted Patient Health Record Representation and Risk Prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 261–269.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[18] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*. 2204–2212.

[19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. *arXiv preprint* (2017).

[20] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 506–519.

[22] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 372–387.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[24] T. Pham, T. Tran, D. Phung, and S. Venkatesh. 2017. Predicting healthcare trajectories from medical records: A deep learning approach. *J. Biomed. Inform.* (2017).

[25] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks* 12, 1 (1999), 145–151.

[26] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. 2018. Identify Susceptible Locations in Medical Records via Adversarial Attacks on Deep Predictive Models. *arXiv preprint arXiv:1802.04822* (2018).

[27] Rajakrishnan Vijayakrishnan, Steven R Steinhubl, Kenney Ng, Jimeng Sun, Roy J Byrd, Zahra Daar, Brent A Williams, Shahram Ebadollahi, Walter F Stewart, et al. 2014. Prevalence of heart failure signs and symptoms in a large primary care population identified through the use of text and data mining of the electronic health record. *J. Card. Fail.* 20, 7 (2014), 459–464.

[28] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[29] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342* (2017).