MedAttacker: Exploring Black-Box Adversarial Attacks on Risk Prediction Models in Healthcare

Muchao Ye* Junyu Luo† Guanjie Zheng‡ Cao Xiao‡ Ting Wang*

Abstract

Deep neural networks (DNNs) have been broadly adopted in health risk prediction to provide healthcare diagnoses and treatments. To evaluate their robustness, existing research conducts adversarial attacks in the white/gray-box setting where model parameters are accessible. However, a more realistic black-box adversarial attack is ignored even though most real-world models are trained with private data and released as black-box services on the cloud. To fill this gap, we propose the first black-box adversarial attack method against health risk prediction models named MedAttacker to investigate their vulnerability. MedAttacker addresses the challenges brought by EHR data via two steps: hierarchical position selection which selects the attacked positions in a reinforcement learning (RL) framework and substitute selection which identifies substitute with a score-based principle. Particularly, by considering the temporal context inside EHRs, it initializes its RL position selection policy by using the contribution score of each visit and the saliency score of each code, which can be well integrated with the deterministic substitute selection process decided by the score changes. In experiments, MedAttacker consistently achieves the highest average success rate and even outperforms a recent white-box EHR adversarial attack technique in certain cases when attacking three advanced health risk prediction models in the black-box setting across multiple real-world datasets. In addition, based on the experiment results we include a discussion on defending EHR adversarial attacks.

1 Introduction

The increasingly accumulated electronic health records (EHR) data have advanced the field of health analytics, especially the health risk prediction [18, 8, 20, 17] task, which aims to predict future health status of patients according to their historical EHR data and nowadays is commonly conducted by deep neural networks (DNNs). Although the robustness of DNN-based health risk prediction systems is of great importance due to its relation with medical resources planning and human lives, the research on adversarial attack which quantifies the robustness of DNNs by generating adversarial examples to fool the victim models has not received as much attention on EHR data as on image or text data. Specifically, there is a gap between the settings of existing adversarial attack approaches and real-world application scenarios on health risk prediction models.

To illustrate, existing studies mainly explore the robustness of deep health risk prediction models by white/gray-box adversarial attacks, which assume attackers can access the parameters of health risk prediction models. For example, Sun et al. [21] propose a white-box one to identify susceptible locations in clinical time series data, An et al. [1] generate adversarial EHR examples in the white/gray-box setting, and Wang et al. [23] test a white-box evasion attack on EHRs. However, in the real world health analytics companies train their models with their private data and release them as black-box services on the cloud. Therefore, the assumptions of white/gray-box settings are often invalid in real-world practice because the parameters of proprietary models of companies are inaccessible. Thus, it is desirable to have a black-box adversarial attack method for understanding the robustness of deep health risk prediction models, but it is difficult to design one owing to the following challenges.

Challenges. The challenges stems from the unique structure of EHR data. As shown in Figure 1 compared to text data which is also discrete, EHR is different in its unordered diagnosis codes within each visit and the incidental nature of hospital visits. That is, each longitudinal EHR is a sequence of patient visits...
that show their evolving conditions as well as sporadic incidences, where the diagnosis codes of each visit are unordered. Such uniqueness of EHR raises the challenge of designing adversarial example generation for EHR. Existing black-box techniques attacking discrete data such as text usually use either score-based methods [19, 11, 16], or reinforcement learning ones [25] to find sensitive words and select replacement words. However, their applicability on generating EHR adversarial example is limited: for one thing, score-based approaches determine replacements directly by the calculated saliency scores, and the generated adversarial examples are likely to be locally optimum; for another, existing reinforcement learning methods [25] have not taken the temporal context into consideration when taking actions. Thus, this technical challenge raises the question of how to alleviate these limitations to design a black-box adversarial attack approach against health risk prediction models that takes the properties of EHRs into consideration.

Our Approach. To solve these challenges, in this paper we propose a new black-box adversarial attack method named MedAttacker to explore the robustness of health risk prediction models. To cope with the unique EHR structure, MedAttacker adopts an approach that bridges score-based and reinforcement learning attacks, which has two steps including hierarchical position selection and substitute selection. To be specific, the first step of MedAttacker is to calculate the contribution score of each visit by taking the temporal context into consideration. Next, it calculates the saliency score of each ICD code within each visit. Using the contribution scores and saliency scores as the initialized policy parameters, it adopts a reinforcement learning framework to select the attacked positions in the EHR to generate globally optimized adversarial examples. After determining the attacked positions, MedAttacker selects the substitute code that can bring the highest score change as the replacement of the original one.

Contributions. To sum up, our contributions are as follows: (1) To the best of our knowledge, we are the first to explore the robustness of health risk prediction models via black-box adversarial attacks. Compared with white/gray-box setting, black-box adversarial attack is more realistic, so our work can better approximate the robustness of real-world health risk prediction models. (2) We propose a new black-box adversarial attack method called MedAttacker, which is motivated to solve the challenges brought by the unique EHR data structure and alleviate the limitations of existing black-box adversarial attack techniques. It attacks the risk prediction models by taking the temporal context into consideration, and it can search better globally optimized adversarial examples by adopting a hybrid framework of reinforcement learning and score-based principles. (3) We compare MedAttacker against state-of-the-art black-box adversarial attack methods in terms of attack success rate across three real-world healthcare datasets. Results show that MedAttacker can generate more adversarial examples on average, and even outperforms white-box model LAVA [1] in the Dementia dataset. With the experimental results, we further include a discussion on the defending against adversarial attacks for health risk prediction models.

2 Related Work

2.1 Adversarial Attacks on EHR Data. The vulnerability problem is a vital issue for the deep health risk prediction models for they are applied in the healthcare domain. Thus, we should understand how reliable health risk prediction DNNs are by exploring their robustness against adversarial attacks. However, adversarial attack research on EHR data is still in the early stage, and much more work is needed in designing effective adversarial attack methods in this domain, especially the black-box ones. To our knowledge, Sun et al. [21] and An et al. [1] are the ones who have explored the area of attacking DNNs with white/gray-box methods in the healthcare domain. [21] proposes a white-box adversarial attack method for the EHR data that are described by continues values including vital signs and lab measurements, while [1] conducts white-box and gray-box adversarial attacks on the ICD-based EHR data. Besides, Wang et al. [23] test an orthogonal matching pursuit-guided method for white-box evasion attack on the discrete EHR data.

Nonetheless, the early work neglects the black-box adversarial attack setting, which is more realistic and challenging. Compared to white-box ones, black-box adversarial attack does not allow the attack methods to use the gradient information for adversarial example generation. Such a restriction makes the adversarial attack more difficult, and it is closer to the real-world scenario because the model gradient is unavailable in most cases. In this paper, we propose an effective black-box adversarial attack method which can improve the adversarial attack techniques for health risk prediction DNNs in a more realistic setting.

2.2 Black-box Adversarial Attacks on Text. With the wide utilization of DNNs in various applications, people have become concerned about the reliability and vulnerability of DNNs in different fields. Among the existing adversarial attack work on different kinds of data including graphs [9] and images [12, 14, 6, 5, 15].
text data \cite{11, 16, 19, 25, 10} are the most relative ones to the EHR data because the search space of EHR and text data are both discrete. Thus, black-box text adversarial attack methods can be used as baselines in our experiments, including DeepWordBug \cite{11}, TextBugger \cite{16}, PWWS \cite{19} and a reinforcement learning method \cite{25}. Among them, DeepWordBug, TextBugger and PWWS can be categorized into score-based methods, which determine the attacked positions and perturbations by the saliency scores and substitute score changes, and \cite{25} proposes a reinforcement learning technique that can dynamically try different kinds of substitute combinations.

As for our work, MedAttacker aggregates the temporal context into a reinforcement learning framework to make it fit for EHR data, and it can be regarded as a hybrid method of the score-based and the reinforcement learning ones. Our experiments will show that such a hybrid design is a more effective adversarial attack solution for health risk prediction models in the black-box setting.

3 Methodology

3.1 Problem Definition

Definition 1 (Electronic Health Records). In our work, the EHRs of all patients are encoded by a high dimensional dictionary called ICD-9, \footnote{https://www.cdc.gov/nchs/icd/icd9.htm} (International Classification of Diseases, Ninth Revision) and each symptom or abnormal finding is encoded into a unique code. In other words, each ICD-9 code is like a discrete symbol of the ICD-9 dictionary as an abstract of a unique medical symptom. Mathematically, for a specific patient whose EHRs are denoted as $\mathbf{V}$, $\mathbf{V}$ is in the form of $[v_1, v_2, \cdots, v_T]$, where $v_t$ ($1 \leq t \leq T$) represents the result of visit $t$, and $T$ is the total number of visits. Each individual visit $v_t = [c_1, c_2, \cdots, c_{n_t}]$ includes $n_t$ diagnosis codes encoded by the ICD-9 system.

Problem 1 (EHR Adversarial Attack). Let $F$ denote the health risk prediction DNN model. Given the input $\mathbf{V}$ of the patient and the corresponding ground truth label $y \in \mathcal{Y} = \{0, 1\}$, where $y = 1$ represents that patient will suffer from the target disease as a positive case and a negative one otherwise, in the training phase health risk prediction model $F$ is trained to generate a prediction score $\mu$ that is as close as to $y$, i.e., $\mu = F(\mathbf{V})$. Suppose that we have a test sample $\mathbf{V}_{\text{test}}$ whose ground truth label is $y_{\text{test}}$, and the classification threshold $\delta$. The output label given by $F$ will be $\hat{y} = \text{sgn}(F(\mathbf{V}_{\text{test}}) - \delta)$, where $\text{sgn}(x) = 1$ if $x > 0$ and $\text{sgn}(x) = 0$ otherwise.

If the victim model $F$ can correctly predict the future status of the patient given the sample $\mathbf{V}_{\text{test}}$, i.e., $\hat{y} = y_{\text{test}}$, the target of adversarial attack is adding a perturbation $\Delta \mathbf{V}_{\text{test}}$ to construct the adversarial example $\mathbf{V}'_{\text{test}} = \mathbf{V}_{\text{test}} + \Delta \mathbf{V}_{\text{test}}$ such that $\hat{y}' = \text{sgn}(F(\mathbf{V}'_{\text{test}}) - \delta) \neq y_{\text{test}}$, where the perturbation $\Delta \mathbf{V}_{\text{test}}$ should be as small as possible and is restricted by $||\Delta \mathbf{V}_{\text{test}}|| < \epsilon$. We denote $||\Delta \mathbf{V}_{\text{test}}||$ as the number of diagnosis code changes because EHR data is in a discrete space and $\epsilon$ as the maximum allowed attacks. We restrict the adversarial attack operation as substitution in our work because addition and deletion can be regarded substitutions related to a null character.

3.2 Proposed Method. As shown in Figure 2, the proposed MedAttacker for adversarial EHR example generation in the black-box setting includes two steps, i.e., hierarchical position selection (selecting the positions of the attacked diagnosis codes by considering temporal context) and substitute selection (selecting the substitutes to replace the attacked diagnosis codes). In the first step, MedAttacker frames the position selection as a policy learned through reinforcement learning (RL). In this formulation, the agent is MedAttacker, the environment consists of the EHR sample $\mathbf{V}$ and victim
model $F$, and the state $s$ is represented by the EHR sample. Suppose it has $M$ learning episode to update the policy parameters, in each episode it will take several steps of actions. Due to the hierarchical characteristics of EHR data, i.e., code $\rightarrow$ visit $\rightarrow$ EHR, $\text{MedAttacker}$ will select the attacked visit firstly and the attacked diagnosis code within the visit later. They are then grouped as the action $a$ taken by the agent, and the policy is parameterized as $\Theta$. To make use of the temporal context, we initialize the policy $\Theta$ by the contribution scores of visits and saliency scores of diagnosis codes. As for the second step, we adopt a score-based fashion to determine the substitute code for the attacked position depending on which is the one that brings the maximum replacement score change in current state, which is integrated into the RL framework and harnessed as the reward $r$ for the agent $\text{MedAttacker}$ to update its policy. The details of these two processes are as follows.

**Hierarchical Position Selection.** The first step to generate an adversarial example is selecting the position of the attacked diagnosis codes. Existing black-box adversarial attack methods \cite{19, 25, 11} on discrete data normally determine the attacked position directly by how much information will lose after removing the attacked word, but they neglect the context of the words within the same sentence, which leads to suboptimal results because existing health risk prediction models treat EHR data in a hierarchical way. Therefore, the position selection process in EHR adversarial attack should be conducted in a hierarchical way: selecting the attacked diagnosis code by firstly selecting the attacked visit and then deciding the attacked position within the selected visit.

Thus, in our RL framework, this action is represented by two sets of parameters, and it is updated by the policy gradient \cite{24} framework. Without the loss of generality, suppose we have a positive test sample $V$ as shown in Figure \ref{diagram} which can be correctly predicted by the trained model $F$, and $V = [v_1, v_2, \ldots, v_T]$ has information of $T$ visits. For the $i$-th visit $v_t$, it has $n_t$ codes. Thus, the parameters to be learned include $p_v = [p_1^{(v)}, \ldots, p_T^{(v)}]$, which is the probability distribution of selecting the visit position, and a group of parameters $p_c = \{p_1^{(c)}, \ldots, p_T^{(c)}\}$, which is the probability distribution of selecting the code position when the visit position is determined. For each $p_t^{(c)} \in p_c$, $p_t^{(c)} = [p_1^{(t)}, \ldots, p_{n_t}^{(t)}]$, where $n_t$ is the number of codes in the $t$-th visit. Thus, the policy parameters to be learned in the policy gradient framework are $\Theta = \{p_v, p_c\}$. In a learning episode, $\text{MedAttacker}$ selects the attacked visit $v_t$ by sampling from $p_v$, and it then decides the attacked position by sampling from $p_t^{(c)}$.

The reason that we adopt a RL framework to select the attacked position is that it enables the adversarial example generation to be a stochastic process instead of a deterministic one, which can allow us to approximate the globally optimized adversarial example with more choices and give us more chances to successfully fool the trained model $F$. The difference between our framework and existing black-box RL framework \cite{25} is that ours will fit better with EHRs by selecting substitutes through replacement score change and taking the temporal context into consideration, which will be detailed as follows.

**Substitute Selection.** After we sample from the policy parameters $\Theta$ and get the position that we are going to attack, the next step is to select a substitute to replace the attacked diagnosis code and generate the adversarial example. Suppose that the code to be attacked is $c_i$ from visit $v_t$, and we denote $S$ as the set of substitute codes.\footnote{We define set $S$ as the set of codes in the same ICD-9 category of $c_i$ as the semantic constraints.} We then use the score changes brought by the substitutes to determine the substitute $c'_i$ for $c_i$. That is, for each substitute code $c \in S$, we can calculate the replacement score change by Eq. \eqref{eq:replace_score_change},

$$\Delta \mu_{t}^{(i)} = F((V - c_i) \cup c) - F(V),$$

where $((V - c_i) \cup c)$ represents the EHR sample where $c_i$ is removed from $V$ and replaced by $c$ in the attacked position. After obtaining all $\Delta \mu_{t}^{(i)}$ scores for every $c$ in set $S$, we determine the best substitute code as demonstrated by Eq. \eqref{eq:best_substitute_code},

$$c'_i = \arg \max_{c \in S} \Delta \mu_{t}^{(i)},$$

where $c'_i$ is the code that we will finally employ to replace the attacked diagnosis code $c_i$.

In the model design, we determine the substitute code by Eq. \eqref{eq:best_substitute_code} instead of sampling by RL for it will be difficult to only use the max $\Delta \mu_{t}^{(i)}$ to update position selection and substitute selection policy parameters simultaneously. We will have a detailed discussion on the policy update later.

**Policy Update.** Given the framework above, the final design problem is how to update the parameters $\Theta$, which can be further divided into the sub-problems of how to initialize $\Theta$ and how to define the rewards.

**Parameter Initialization.** Existing RL based black-box adversarial attack methods \cite{25} choose uniform distribution for initialization, but we find that using the \textit{contribution scores} and \textit{saliency scores} is
a better way to initialize the policy $\Theta$ for it takes temporal context into consideration. Specifically, we utilize the visit contribution scores to initialize $p_v$ and code saliency scores to initialize $p_c$.

We define the visit contribution score as follows. To calculate the visit contribution of $v_t$, we first calculate the output score given by the trained model $F$ when input is $[v_1, ..., v_t]$ and then calculate the output score when the input is $[v_1, ..., v_{t-1}]$. Next, we compute the contribution score $\xi_t$ by their difference,

$$\xi_t = F([v_1, ..., v_t]) - F([v_1, ..., v_{t-1}]),$$

where $\xi_t$ indicate how much information that the whole visit of $v_t$ can contribute to improving the health risk prediction given the context $[v_1, ..., v_{t-1}]$. By initializing $p_v$ as the normalized $[\xi_1, ..., \xi_T]$, the temporal context is utilized for determining the attacked position without hurting the stochastic property.

In terms of the saliency score of each code, it is calculated in a similar way as Eq. (3.4). For the $i$-th code $c_i$ in visit $v_t$, we calculate the saliency score as

$$\xi^{(i)}_t = F(V) - F(V - c_i),$$

where $(V - c_i)$ denotes the incomplete EHR data where code $c_i$ is removed. The score $\xi^{(i)}_t$ is used to indicate the information that code $c_i$ possesses for health risk prediction. If score $\xi^{(i)}_t$ is high, it indicates that attacking $c_i$ can bring more salient influence. Thus, we initialize each $p^{(c)}_v$ as the normalized $[\xi^{(1)}_t, ..., \xi^{(n)}_t]$, which fits with the unordered property of EHRs.

**Reward Calculation.** We now discuss how to calculate reward in each learning episode. Our solution is utilizing the maximum replacement score change $\max \Delta \mu^{(i)}_t$ in Eq. (3.3) as the reward $r$ to update the policy parameters $\Theta$, which enables us to integrate the position selection and substitute selection together and help MedAttacker effectively find out the positions useful for adversarial example generation. Thus, in each learning episode, the total rewards of the adversarial example generation process is $J(\Theta) = E(\sum_{t=0}^{T-1} \gamma^t r_t(\Theta))$, where $r_t$ is the reward attained in the step $t$, and $\gamma \in [0, 1]$ is the discount factor set to be 0.95.

In addition, we update $\Theta$ by the policy gradient method, in which the gradient of $J(\Theta)$ can be approximated by the REINFORCE algorithm [24], and the gradient $\nabla_\Theta J(\Theta) = \sum_{t=0}^{T-1} \nabla_\Theta \log \pi_\Theta(\alpha_t | s_t) G_t$, where $\alpha_t$ is the action that MedAttacker takes in step $t$, $s_t$ is the state of the environment, and $\pi_\Theta(\alpha_t | s_t)$ is the probability of taking action $\alpha_t$ in state $s_t$, and $G_t$ refers to the discounted future reward $\sum_{t'=t}^{T-1} \gamma^{t-t'} r_{t'}$. Given the approximated gradients, we can update the parameters according to $\Theta \leftarrow \Theta + \alpha \nabla_\Theta J(\Theta)$, where $\alpha$ is the learning rate.

### 4 Experiments

In this section we will show the experimental results for evaluating the effectiveness of the proposed method.

#### 4.1 Experimental Setup

**4.1.1 Datasets.** In our experiments, we use three real-world health insurance claim datasets, including heart failure, kidney disease, and dementia, which are collected by a health information technology company. The statistics of these datasets are shown in Table 1. All the black-box adversarial attack experiments are conducted on the test data. The average number of visits on the heart failure, kidney disease, and dementia datasets is 38.74, 39.09, and 41.05, respectively. The average number of ICD codes per visit is 4.24, 4.40, and 4.71 on the heart failure, kidney disease, and dementia datasets, respectively.

| Dataset          | Heart Failure | Kidney Disease | Dementia |
|------------------|---------------|----------------|----------|
| Total Cases      | 12,320        | 11,240         | 9,540    |
| Positive Cases   | 3,080         | 2,810          | 2,385    |
| Negative Cases   | 9,240         | 8,430          | 7,155    |
| Test Set Size    | 1,848         | 1,686          | 1,431    |
| Unique ICD-9 Codes| 8,692        | 8,802          | 7,813    |

#### 4.1.2 Victim Models.** Since we are conducting adversarial attacks on EHR data, we select three representative DNNs designed for health risk prediction task as the victim models in the adversarial attacks, which are Retain [8], SAnD [20] and HiTANet [17]. The reason for selecting them is that Retain and SAnD are two state-of-the-arts that employ two mostly used temporal models in deep learning, i.e., recurrent neural networks (RNNs) and Transformer [22], and HiTANet is a method that emphasizes the utilization of time information, which is widely used [2 3 17] in risk prediction.

**4.1.3 Baselines.** Since we are the first to work on the adversarial attack on EHRs, there is few baselines specifically designed for the EHR adversarial attack. Therefore, most baselines that are used in the experiments are originally designed for the text adversarial attack. To make the input data fit the text adversarial attack models, we treat each visit as a sentence and treat each code as a word. In our experiments, we use six baselines, including a naive approach and five (including four score-based and one reinforcement learning-based) state-of-the-art black-box adversarial attack methods as follows:

1. **Random**, a naive adversarial attack baseline which randomly selects the attacked positions and substitutes;
2. **DeepWordBug** [11];
3. **TextFooler** [3];
4. **MedFool** [2];
5. **WordBug** [2];
6. **Saliency 1** [17].

#### 4.2 Experimental Results

The statistics of the used datasets.

| Dataset          | Heart Failure | Kidney Disease | Dementia |
|------------------|---------------|----------------|----------|
| Total Cases      | 12,320        | 11,240         | 9,540    |
| Positive Cases   | 3,080         | 2,810          | 2,385    |
| Negative Cases   | 9,240         | 8,430          | 7,155    |
| Test Set Size    | 1,848         | 1,686          | 1,431    |
| Unique ICD-9 Codes| 8,692        | 8,802          | 7,813    |
Table 2: Comparison on the number of successful attacks (the first row in each block) and success rate (the second row in each block). The average success rate is calculated for comparison when victim model is unknown.

| Dataset            | Heart Failure | Kidney Disease | Dementia |
|--------------------|---------------|----------------|----------|
|                    | HITANet       | Retain SANd    | Average  |
| Random             | 30 (1.62%)    | 18 (0.97%)    | 7 (0.99%) |
|                    | 21 (1.25%)    | 10 (0.59%)    | 32 (1.90%)|
|                    | 21 (1.68%)    | 18 (1.20%)    | 10 (0.70%)|
| TextBugger         | 216 (11.69%)  | 119 (6.64%)   | 4 (2.22%) |
|                    | 182 (10.79%)  | 138 (6.17%)   | 104 (6.17%)|
|                    | 117 (8.18%)   | 109 (7.62%)   | 6 (0.42%) |
| DeepWordBug        | 231 (12.50%)  | 113 (6.11%)   | 12 (0.65%)|
|                    | 248 (14.71%)  | 96 (5.69%)    | 92 (5.46%)|
|                    | 114 (8.18%)   | 87 (7.26%)    | 15 (0.42%)|
| PWWS-Saliency      | 277 (14.99%)  | 129 (6.98%)   | 48 (2.60%)|
|                    | 264 (15.66%)  | 98 (5.81%)    | 229 (13.58%)|
|                    | 249 (19.36%)  | 106 (11.60%)  | 66 (4.61%)|
| PWWS               | 369 (19.97%)  | 102 (8.77%)   | 52 (2.81%)|
|                    | 332 (19.69%)  | 154 (9.13%)   | 239 (14.18%)|
|                    | 239 (19.36%)  | 204 (14.26%)  | 77 (5.38%)|
| RL-Attack          | 349 (18.78%)  | 146 (7.90%)   | 25 (1.35%)|
|                    | 301 (17.85%)  | 132 (7.83%)   | 142 (8.42%)|
|                    | 271 (19.01%)  | 160 (11.18%)  | 30 (2.10%)|
| MedAttacker        | 426 (10.05%)  | 166 (8.08%)   | 64 (3.46%)|
|                    | 369 (21.89%)  | 149 (8.84%)   | 218 (12.95%)|
|                    | 349 (21.89%)  | 218 (12.95%)  | 218 (12.95%)|

TextBugger [16]; (4) PWWS [19] and its variant (5) PWWS-Saliency which only uses the saliency score to determine which word to be attacked; and (6) RL-Attack [25]. In addition, we use the state-of-the-art white-box EHR adversarial attack method LAVA [1] which can access the model gradients as a baseline to better understand the attack effect of black-box adversarial attacks.

4.1.4 Implementation. Our experiments are implemented in the PyTorch framework in the hardware environment of a NVIDIA Tesla P100 GPU and Intel Xeon E5-2680 CPUs. The reinforcement learning environment is implemented in the OpenAI Gym [3] package, and the learning rate of the policy parameters is $1 \times 10^{-3}$. When implementing our algorithm, we set the hyperparameter $l = 500$. The set of $S$ for each code is made up of diagnosis codes in the same ICD category by the Clinical Classification Software-DIAGNOSES. We set size of $S$ no more than 10 for efficiency reason and for each category they are selected randomly. As for baselines, the substitute selection of the score-based methods are set as the same way that PWWS does for fair comparison, which is decided by the $\Delta P$ score. We vary the the maximum allowed attacks $\epsilon$ from 5 to 15.

4.1.5 Evaluation Metrics. The main goal of attacking victim models is to change the labels of test set data that are correctly predicted by the black-box DNNs, so the first metric is the number of successful attacks that each method make, and dividing it by the size of test set can get the success rate [25] [10], which shows the covering range of successful attacks and the decreased accuracy of the victim model. We can then calculate the average success rate for one dataset across different health risk prediction models to estimate the attack result when the attacked model is unknown as a black-box. The larger they are, the better the performance is.

4.2 Performance Evaluation. Since there is always physical restriction on the access of EHR data in real-world defense, we first validate the performance of models under a relatively restrict and realistic setting: (1) The maximum accessible visits allowed for attack in the test set is 20, which is about a half of average numbers of visits of three datasets, and (2) the maximum allowed attacks $\epsilon = 5$. Such a setting can better validate the effectiveness of designed method for testing the robustness of health risk prediction models. The comparison of adversarial attack results between MedAttacker and baselines are shown in Table 2. From Table 2 we can see that MedAttacker can always attain the highest average success rate across three test datasets, which is $11.83\%$, $14.55\%$ and $15.30\%$ in the heart failure, kidney disease and dementia dataset, respectively. Specifically, it achieves the best performance over three datasets against three different victim models in 6 out of 9 cases. These results demonstrate that MedAttacker has the best generalization ability for EHR adversarial attack compared to existing black-box adversarial attack techniques, and such a versatile attack ability indicates that MedAttacker can be well applied in the black-box setting when victim models are unknown.

Besides, MedAttacker is especially good at dealing with difficult situations when attacking in relatively large datasets or against victim models using time information. Among all the three datasets, MedAttacker always has the highest success rate in the largest data, i.e., heart failure, against different victim models. Compared to the other two datasets, the test samples in heart failure are more varied and diverse. Thus, in this complex situation, MedAttacker is able to bring out the potential of reinforcement learning and generate
Table 3: Comparison with LAVA on success rate.

| Victim Model | Retain |
|--------------|--------|
| Method       | Heart Failure | Kidney | Dementia |
| LAVA         | 12.34%    | 10.85% | 11.39%    |
| MedAttacker  | 8.98%     | 8.84%  | 14.68%    |

more adversarial examples to successfully fool the victim models. Moreover, MedAttacker can constantly achieve the best attack success rate when the victim model is HiTANet, which employs time information for health risk prediction, and the success rate is 3.08%, 2.20%, and 1.74% higher than the second best method PWWS in heart failure, kidney disease and dementia dataset, respectively. This is mainly because the hierarchical position selection strategy is useful for MedAttacker to attack health risk prediction models using time information by discovering most contributory visits for time embedding learning, which is an advanced design [2, 3, 17] widely used in health risk prediction.

Discussion. The results above show that MedAttacker can alleviate the limitations of reinforcement learning and score-based principles to have better generalization ability while conducting attacks in complex situations. Compared to score-based methods [11, 16, 19], MedAttacker selects attacked code positions in a stochastic fashion by reinforcement learning, which can help MedAttacker generate more effective adversarial examples to fool the victim models. In addition, compared to RL-Attack [25], MedAttacker initializes its parameters to take the temporal context into consideration and is only required to learn the position selection parameters, which can help it find out better substitutes for adversarial attack. Hence, MedAttacker is more suitable for EHR adversarial attacks.

4.3 Comparison with White-box Attack. To evaluate the effectiveness of black-box adversarial attacks on EHRs, we compare the attack results between black-box adversarial attacks and white-box ones. Thus, we also employ LAVA as a baseline for white-box adversarial attack, and the experimental results are listed in Table 3. We compare them in the case of Retain owing to the availability of official implementation codes of LAVA. Because white-box ones have the knowledge of gradients, it is not surprising to see that LAVA has better performance on the datasets of heart failure and kidney disease. But compared to LAVA, MedAttacker can still have 72.77% and 81.47% adversarial attack effects of the white-box method on heart failure and kidney disease datasets, respectively, which demonstrates the effectiveness of the designed method.

Most importantly, MedAttacker can have better attack results on the dementia dataset. This indicates that for a relatively small dataset such as the dementia one, black-box adversarial attack techniques have the potential to achieve better results against white-box ones because the size of the dataset may restrict the model parameter learning, which may mislead the white-box attack result owing to inaccurate gradients.

4.4 Model Analysis. Without the loss of generality, we conduct three groups of model analysis experiments on the heart failure disease against HiTANet.

4.4.1 Ablation Study. To validate the module design, as shown in Table 4 we utilize three variants of MedAttacker for comparison in ablation study experiment. From Table 4 we can find that if MedAttacker selects substitutes stochastically (“w/o Substitute Selection”) as RL-Attack does, the performance will decrease by 5.03%, which shows that we can release its limitation by selecting substitute in a score-based fashion as Eq. (3.3) indicates. As for the position selection strategy, if MedAttacker does not select the attacked position in a hierarchical way (“w/o Hierarchical Position Selection”), the attack effect on success rate will decrease by 2.22%. Hence, the hierarchical position selection is more suitable for the EHR structure. In addition, if MedAttacker initializes its policy uniformly (“w/o Score-based Initialization”), its performance will decrease from 23.05% to 22.19%. Thus, we should initialize the policy parameters by the contribution and saliency scores for it aggregates position selection with the temporal context.

4.4.2 Scalability Study. To validate the model scalability, we conduct an experiment by changing the maximum allowed attacks from 5, 10 to 15, while keeping the maximum accessible visits as 20. The experimental results are shown in Figure 3(a). From the results we can see that if we allow MedAttacker to conduct more attacks on the victim model, the accuracy will keep increasing from 23.05% to 30.52%, which satis-

Table 4: Model analysis on module design.

| Victim Model | HiTANet |
|--------------|---------|
| Modules      | Success Rate |
| w/o Substitute Selection | 18.02% |
| w/o Hierarchical Position Selection | 20.83% |
| w/o Score-based Initialization | 22.19% |
| MedAttacker  | 23.05%  |
4.6 Discussion on Health Risk Prediction Defense. After finding an effective black-box adversarial attack method, based on the experimental results above, we provide the following suggestions on defending against EHR adversarial attack in real-world applications: for one thing, restricting the visibility of EHR data. Since only allowing 5 attacks on 20 visits of the EHR can cause the victim model to decrease its accuracy up to 26.83% by the proposed method, we should design effective mechanism for protecting EHR data. For another, improving health risk prediction model design. We recommend using the Transformer structure as building blocks for the attack success rates caused by MedAttacker in the heart failure and dementia dataset are both less than 5% against SaNd but they are 8.98% and 14.68% against Retain, respectively. This is because Transformer processes the temporal data in parallel, which can reduce the error accumulation problem in RNNs and improve model robustness.

5 Conclusion

The robustness of health risk prediction models is crucial because they are related to human lives. Although researchers have investigated their vulnerability by the white/gray-box adversarial attacks, a more realistic stimulation of real-world adversarial attacks, i.e., EHR adversarial attack in the black-box setting, has not been explored yet. To increase the momentum in this field, we introduce a black-box adversarial attack framework named MedAttacker to explore the robustness of health risk prediction models. It is more suitable for adversarial attack on EHR data in the black-box setting because it takes the temporal context of EHR into consideration, and the stochastic position selection and deterministic substitute selection processes can help it better generate globally optimized adversarial examples. MedAttacker can achieve the highest average success rate in three real-world datasets against three representative health risk prediction models compared with the state-of-the-art baselines. Besides, in
a relatively small EHR dataset, it can even outperform the white-box EHR adversarial attack baseline. Finally, based on our results we include a discussion to help improve the defense mechanism of health risk prediction models.

References

[1] S. An, C. Xiao, W. F. Stewart, and J. Sun, Longitudinal adversarial attack on electronic health records data, in The World Wide Web Conference, 2019, pp. 2558–2564.
[2] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic, Interpretable representation learning for healthcare via capturing disease progression through time, in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 43–51.
[3] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, Patient subtyping via time-aware lstm networks, in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 65–74.
[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, Openai gym, 2016.
[5] N. Carlini and D. A. Wagner, Towards evaluating the robustness of neural networks, in 2017 IEEE Symposium on Security and Privacy, 2017, pp. 39–57.
[6] J. Chen, M. I. Jordan, and M. J. Wainwright, Hopskipjumpattack: A query-efficient decision-based attack, in 2020 IEEE Symposium on Security and Privacy, 2020, pp. 1277–1294.
[7] M. Cheng, T. Le, P. Chen, H. Zhang, J. Yi, and C. Hsieh, Query-efficient hard-label black-box attack: An optimization-based approach, in ICLR, 2019.
[8] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, Retain: An interpretable predictive model for healthcare using reverse time attention mechanism, in NeurIPS, 2016, pp. 3504–3512.
[9] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, Adversarial attack on graph structured data, in International conference on machine learning, PMLR, 2018, pp. 1115–1124.
[10] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, Hotflip: White-box adversarial examples for text classification, in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 31–36.
[11] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, Black-box generation of adversarial text sequences to evade deep learning classifiers, in 2018 IEEE Security and Privacy Workshops, IEEE, 2018, pp. 50–56.
[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, in ICLR, 2015.
[13] D. Holden, T. Komura, and J. Saito, Phase-functioned neural networks for character control, ACM Transactions on Graphics (TOG), 36 (2017), pp. 1–13.
[14] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, Black-box adversarial attacks with limited queries and information, in International Conference on Machine Learning, PMLR, 2018, pp. 2137–2146.
[15] A. Kurakin, I. J. Goodfellow, and S. Bengio, Adversarial examples in the physical world, in ICLR, 2017.
[16] J. Li, S. Ji, T. Du, B. Li, and T. Wang, Textbugger: Generating adversarial text against real-world applications, in 26th Annual Network and Distributed System Security Symposium, 2019.
[17] J. Luo, M. Ye, C. Xiao, and F. Ma, Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 647–656.
[18] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, Kame: Knowledge-based attention model for diagnosis prediction in healthcare, in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 743–752.
[19] S. Ren, Y. Deng, K. He, and W. Che, Generating natural language adversarial examples through probability weighted word saliency, in Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, pp. 1085–1097.
[20] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, Attend and diagnose: Clinical time series analysis using attention models, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.
[21] M. Sun, F. Tang, J. Yi, F. Wang, and J. Zhou, Identify susceptible locations in medical records via adversarial attacks on deep predictive models, in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 793–801.
[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in NeurIPS, 2017, pp. 5998–6008.
[23] Y. Wang, Y. Han, H. Bao, Y. Shen, F. Ma, J. Li, and X. Zhang, Attackability characterization of adversarial evasion attack on discrete data, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1415–1425.
[24] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning, 8 (1992), pp. 229–256.
[25] Y. Zang, B. Hou, F. Qi, Z. Liu, X. Meng, and M. Sun, Learning to attack: Towards textual adversarial attacking in real-world situations, arXiv preprint arXiv:2009.09192, (2020).