Topics in Machine Learning
Machine Learning for Healthcare

Rahul G. Krishnan
Assistant Professor
Computer science & Laboratory Medicine and Pathobiology

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

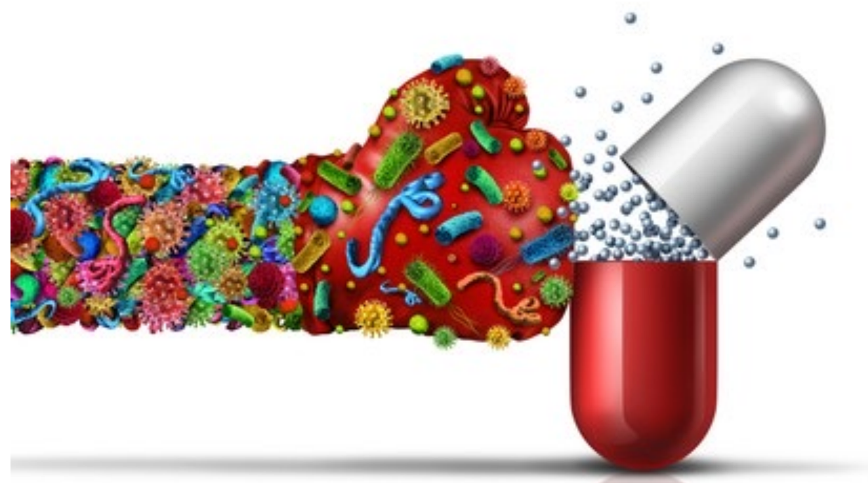
Anncouncements & Outline

- Friday– Nikhil Verma [Model introspection]
- Project report will have a contributions sections:
  - Please list the contributions made by each individual to the report
  - Will not count towards the page limit
- Case studies in machine learning for healthcare
  - C1: Machine learning to reduce antibiotic resistance
  - C2: Adversarial attacks on time-series data in healthcare

Case study 1

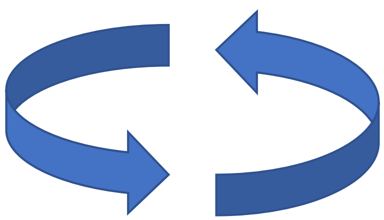
- [A decision algorithm to promote outpatient antimicrobial stewardship for uncomplicated urinary tract infection, Kanjilal et. al, 2020]

Overuse leads to resistance



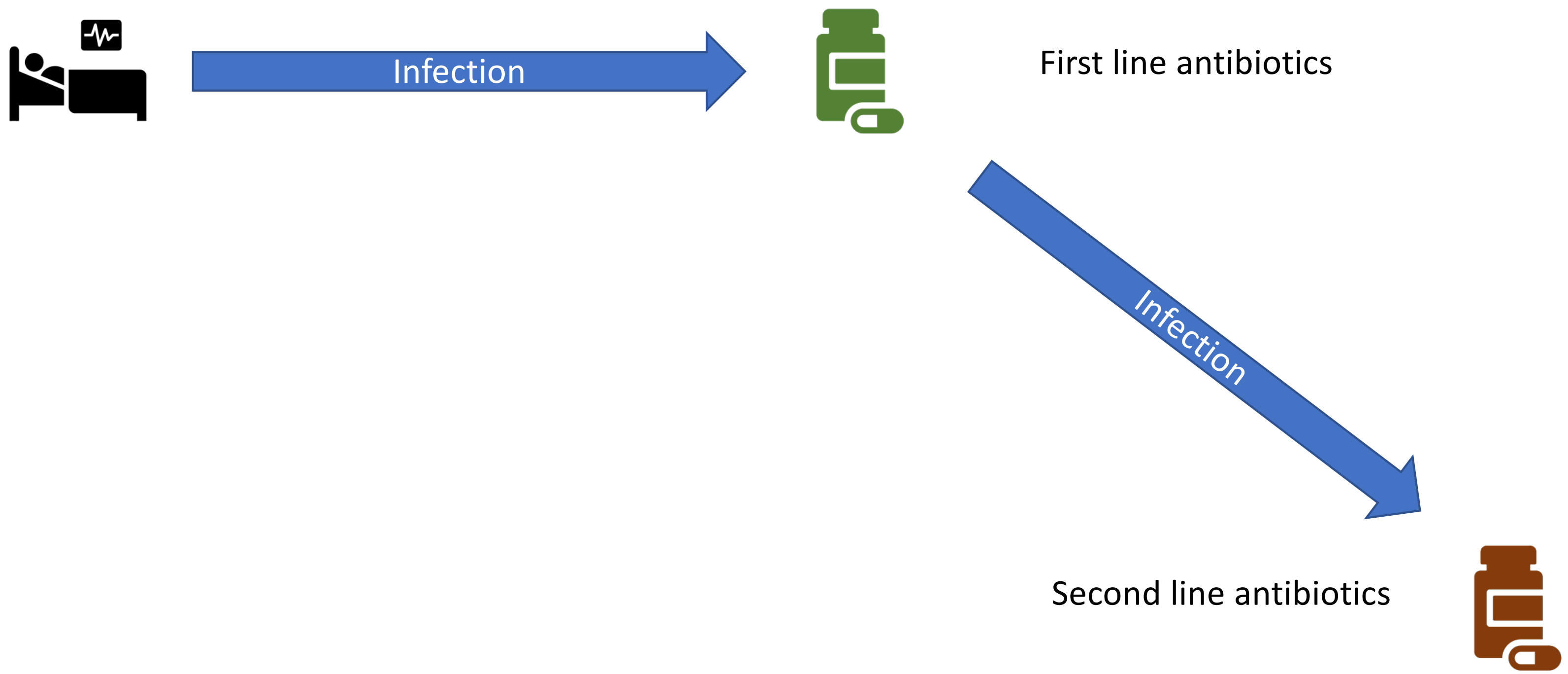
The problem

- The use and over-use of anti-biotics has led to resistance
- Resistance is a cause of treatment failure which triggers further use of broad-spectrum agents which again encourages resistance



- **Disease**: Uncomplicated Urinary Tract Infection in women
  - 13 million outpatient & emergency room visits
  - 4.7 million prescriptions annually

The status quo – (1) the idealized pipeline



The status quo

- Most common prescriptions are : Floroquinolone antibiotics (2^{nd} line)
- Hypothesis is that this is leading to resistance

- What are the clinical regulations:
  - Infectious Disease Society of America (IDSA):
    - Avoid the use of fluoroquinolones
  - Low adherence since end-decision made by clinician

How can we do better?

- Use ML with EHR data to predict likelihood of resistance to first and second line therapy
- Use probabilities to define a decision rule to create recommendations
- Compare recommendations to clinician performance

Cohort

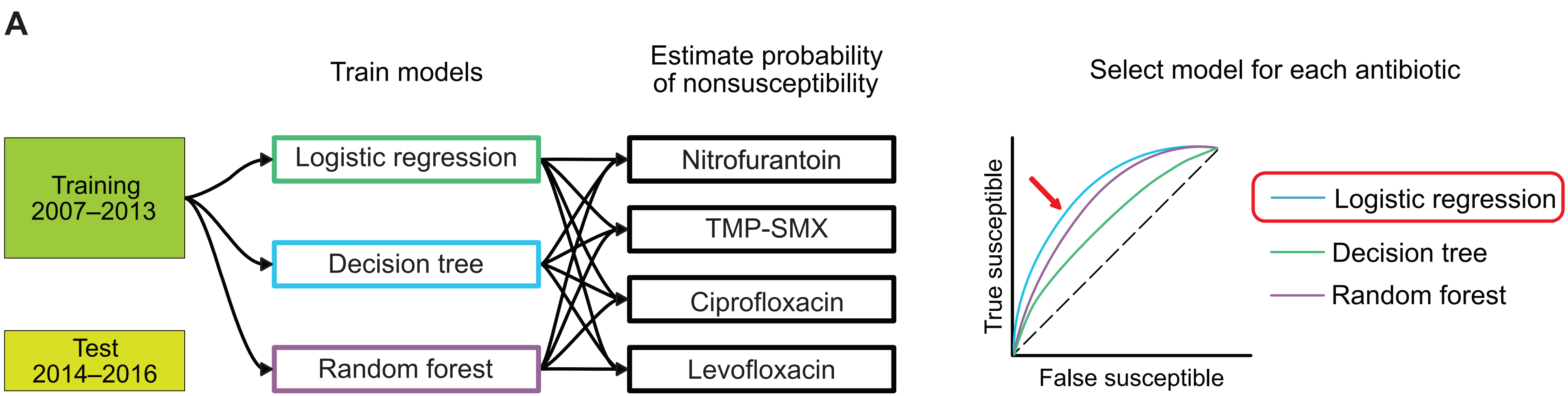
	Entire cohort (2007–2016)
n (patients)	13,682
n (specimens)	15,806
Demographics	
Age, mean (SD)	34.0 (10.9)
Race, *n* (%)	
White	8,784 (64.2)
Non-white	4,898 (35.8)
Location, *n* (%)	
Outpatient	11,639 (85.1)
Emergency room	1,607 (11.7)
General inpatient	534 (3.9)
Intensive care unit	17 (0.1)

What features might you deem relevant for this task?

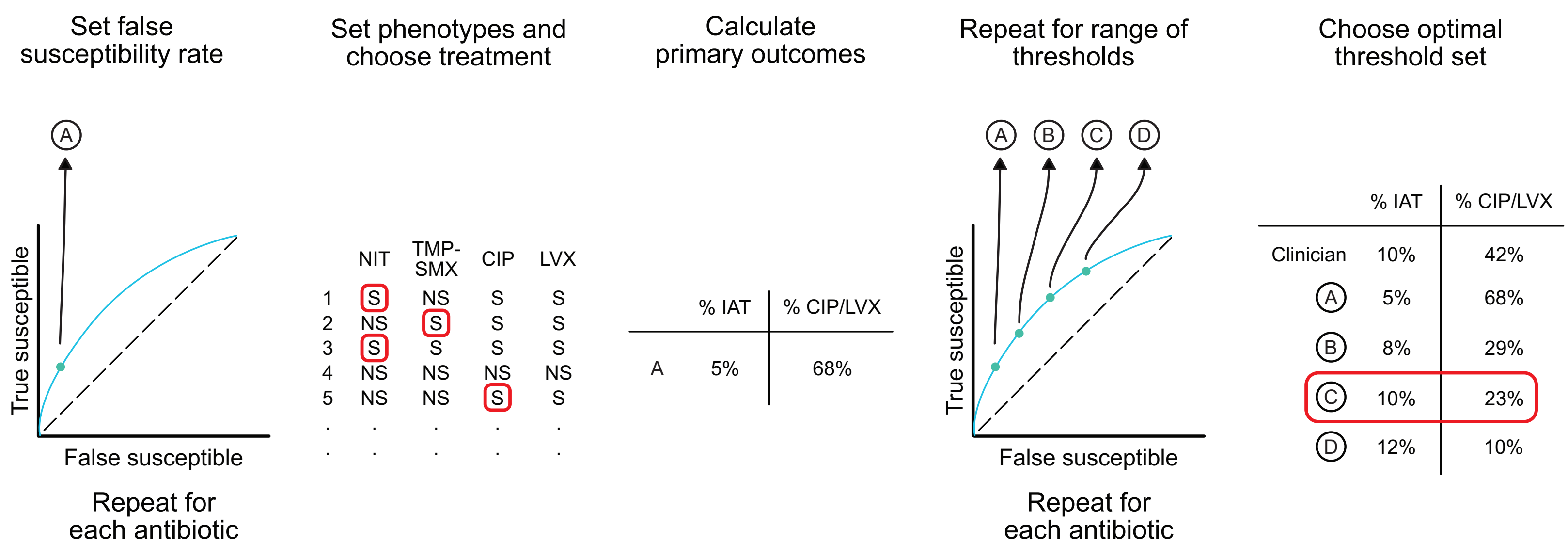
Features

- Demographics
- Microbiology
- Population level prevalence of resistance

Modeling


A
Train models
Estimate probability of nonsusceptibility
Select model for each antibiotic
Training 2007–2013
Test 2014–2016
Logistic regression
Decision tree
Random forest
Nitrofurantoin
TMP-SMX
Ciprofloxacin
Levofloxacin
True susceptible
False susceptible
Logistic regression
Decision tree
Random forest

Pipeline



Found a threshold at which IAT is minimized while second line is set to ~10%

CILP/LVX – proportion of second line therapy

IAT – Inappropriate antibiotic therapy

Comparison to clinicians

Retrain models on entire training dataset

Training
2007–2013

Use optimal thresholds to set phenotypes and choose treatment for **test** isolates

	NIT	TMP-SMX	CIP	LVX
1	NS	NS	NS	NS
2	S	NS	S	S
3	NS	S	NS	NS
4	NS	S	S	S
5	NS	NS	S	S
.	.	.	.	.

Calculate primary outcomes for algorithm, clinicians, and guidelines on **test** data

	% IAT	% CIP/LVX
Guidelines	11%	10%
Clinician	12%	35%
Algorithm	10%	11%

Evaluating what the algorithm would have done in different scenarios

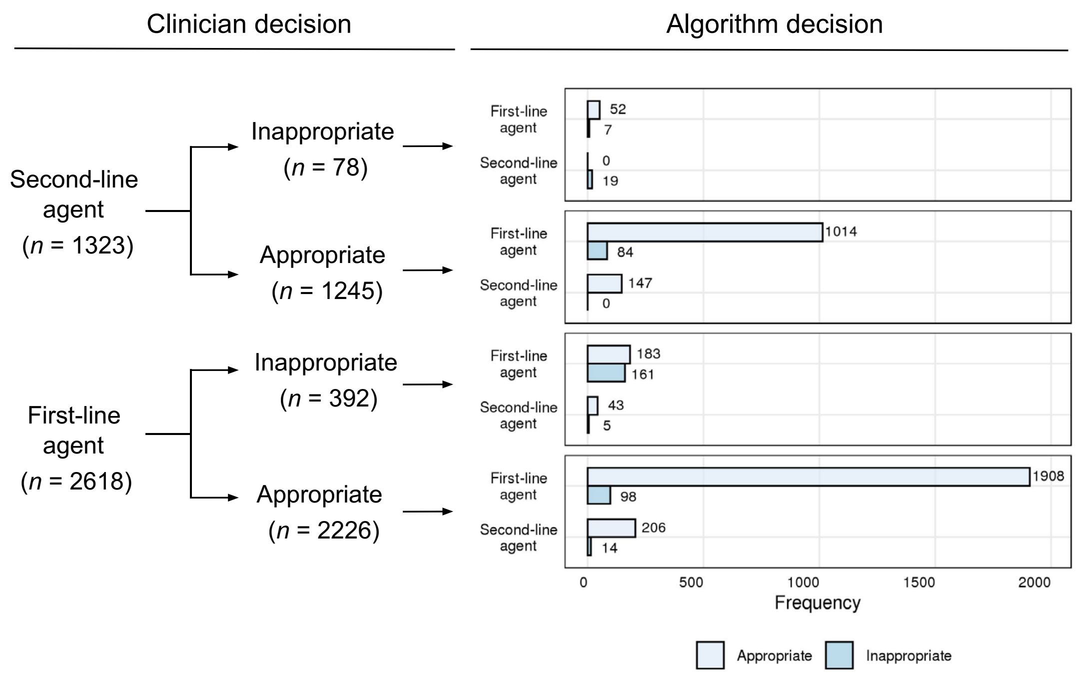


Fig. 4. Post hoc analysis of clinician versus algorithm therapy decisions and appropriateness in patients with uncomplicated UTI presenting between 2014 and 2016. Appropriate therapy was defined as the choice of an empiric antibiotic that has in vitro activity against the pathogen, whereas inappropriate therapy was defined as the choice of an empiric antibiotic that has no in vitro activity against the pathogen.

Conclusion

- Data available to experiment with: https://physionet.org/content/antimicrobial-resistance-uti/1.0.0/

Case study 2

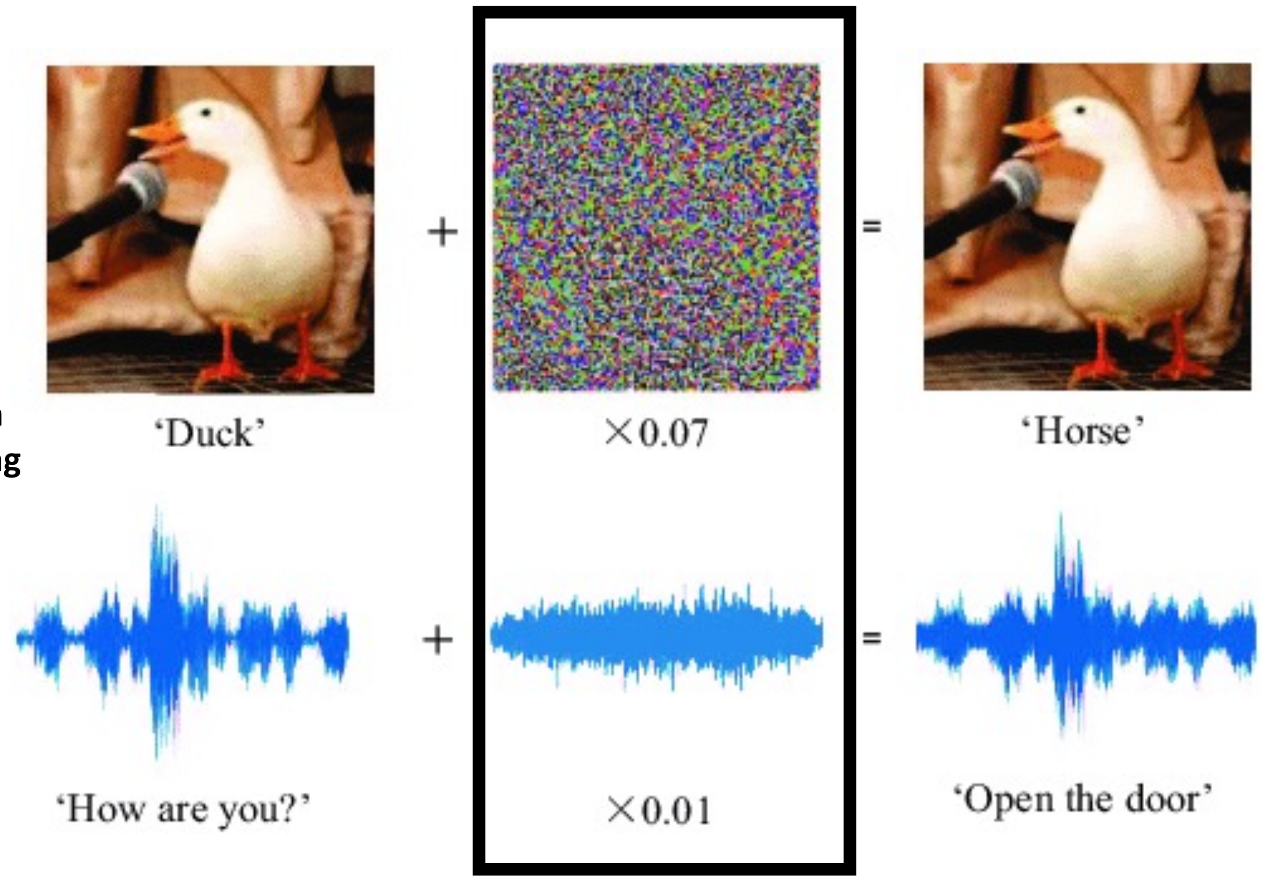
- [Deep learning models for electrocardiograms are susceptible to adversarial attack, Han et. al, 2020]

AliveCor nets $65M, new FDA clearances for future telehealth plans

Source: https://www.fiercebiotech.com/medtech/alivecor-nets-65m-new-fda-clearances-for-future-telehealth-plans

Adversarial examples in machine learning

Model dependent inputs that can change the prediction of any machine learning classifier.



Why might adversarial examples be a problem in healthcare?

(Not necessarily good)use cases for automated ECG prediction

- Lower/higher insurance rates for individuals who undergo regular ECG assessments
- Referrals to specialists based on automated neural network prediction

New adversarial attack for ECG data

- Electrocardiograms:
  - 12 lead ECGs used to assess heart function
  - Apple Watches use a single lead ECG to detect arrythmias
- Contributions of this work:
  - Showcase how to create an adversarial attack for ECG data
- But first, some context on gradients, GradCAM and adversarial attacks

Interpreting linear models

Model	Equation	Interpretation
Level-Level Regression	$Y = \alpha + \beta X$	One unit change in X leads to β unit change in Y
Log-Linear Regression	$log(Y) = \alpha + \beta X$	One unit change in X leads to $100 * \beta$ percent change in Y
Linear-Log Regression	$Y = \alpha + \beta\, log(X)$	One percent change in X leads to $\beta/100$ unit change in Y
Log-Log Regression	$log(Y) = \alpha + \beta\, log(X)$	One percent change in X leads to β percent change in Y

Linear models are inherently interpretable.

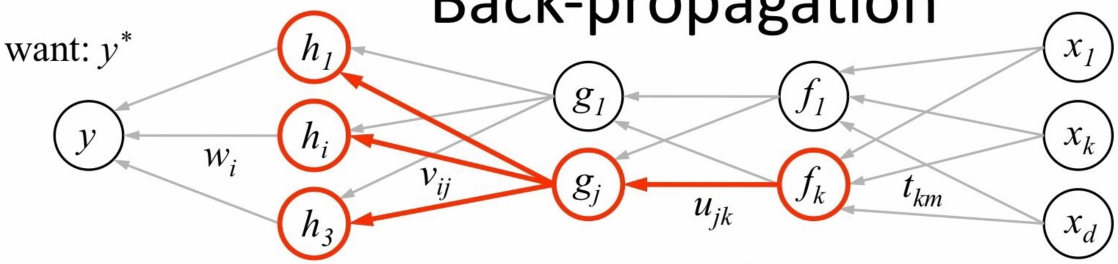
What about non-linear models?

Source: https://www.kdnuggets.com/2017/10/learn-generalized-linear-models-glm-r.html/2

How might you interpret nonlinear models?

Gradients with respect to the input

Back-propagation



1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target y^*
2. **feed forward:** for each unit g_j in each layer 1…L
   compute g_j based on units f_k from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction y and error $(y\text{-}y^*)$
4. **back-propagate error:** for each unit g_j in each layer L…1

(a) compute error on g_j

$$\underbrace{\frac{\partial E}{\partial g_j}}_{\text{should } g_j \text{ be higher or lower?}} = \sum_i \underbrace{\sigma'(h_i) v_{ij}}_{\text{how } h_i \text{ will change as } g_j \text{ changes}} \underbrace{\frac{\partial E}{\partial h_i}}_{\text{was } h_i \text{ too high or too low?}}$$

(b) for each u_{jk} that affects g_j

(i) compute error on u_{jk}

$$\frac{\partial E}{\partial u_{jk}} = \underbrace{\frac{\partial E}{\partial g_j}}_{\text{do we want } g_j \text{ to be higher/lower}} \underbrace{\sigma'(g_j) f_k}_{\text{how } g_j \text{ will change if } u_{jk} \text{ is higher/lower}}$$

(ii) update the weight

$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$

We can use the same algorithm
we use for learning

Gradients wrt inputs - Class Activation Maps

Gradients wrt inputs - Adversarial attacks

FGSM: FGSM is a fast algorithm. For an attack level ε, FGSM sets

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y))$$

By repeatedly changing the datapoint x very slightly, we can change the model's output from turtle to a rifle.

Back to our paper

Adversarial attacks for ECG data

FGSM: FGSM is a fast algorithm. For an attack level ε, FGSM sets

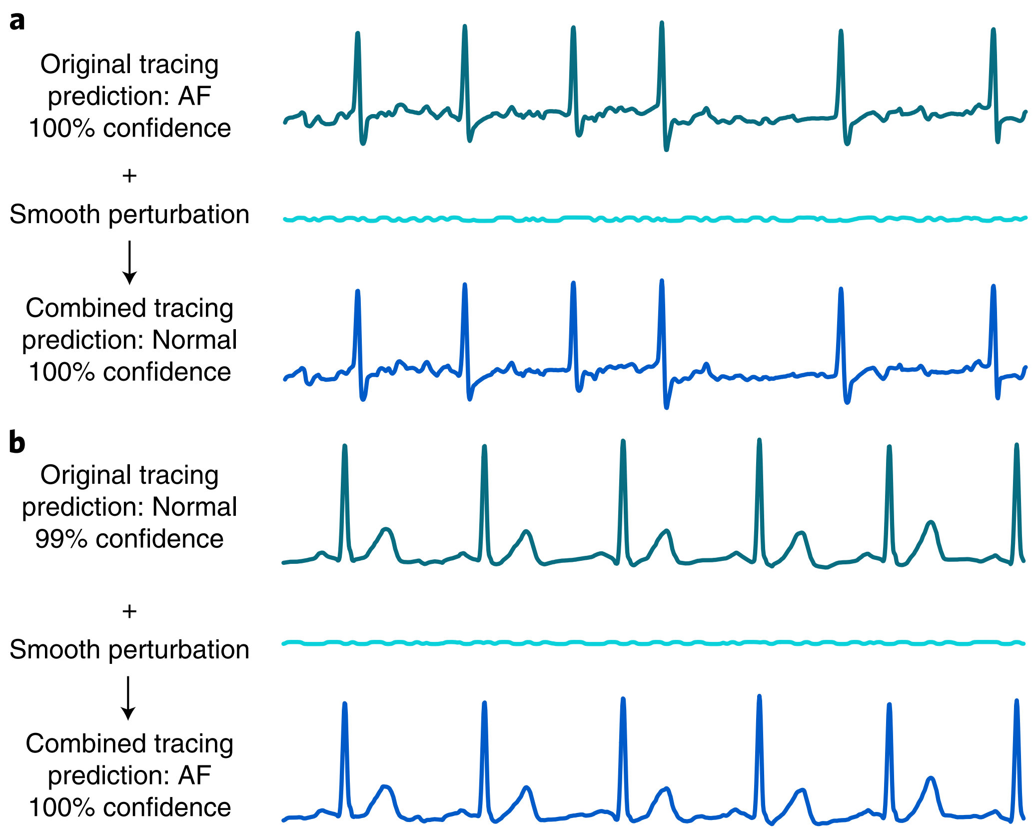
$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y))$$

In this work, they use a slightly different kind of attack called the projected gradient attack

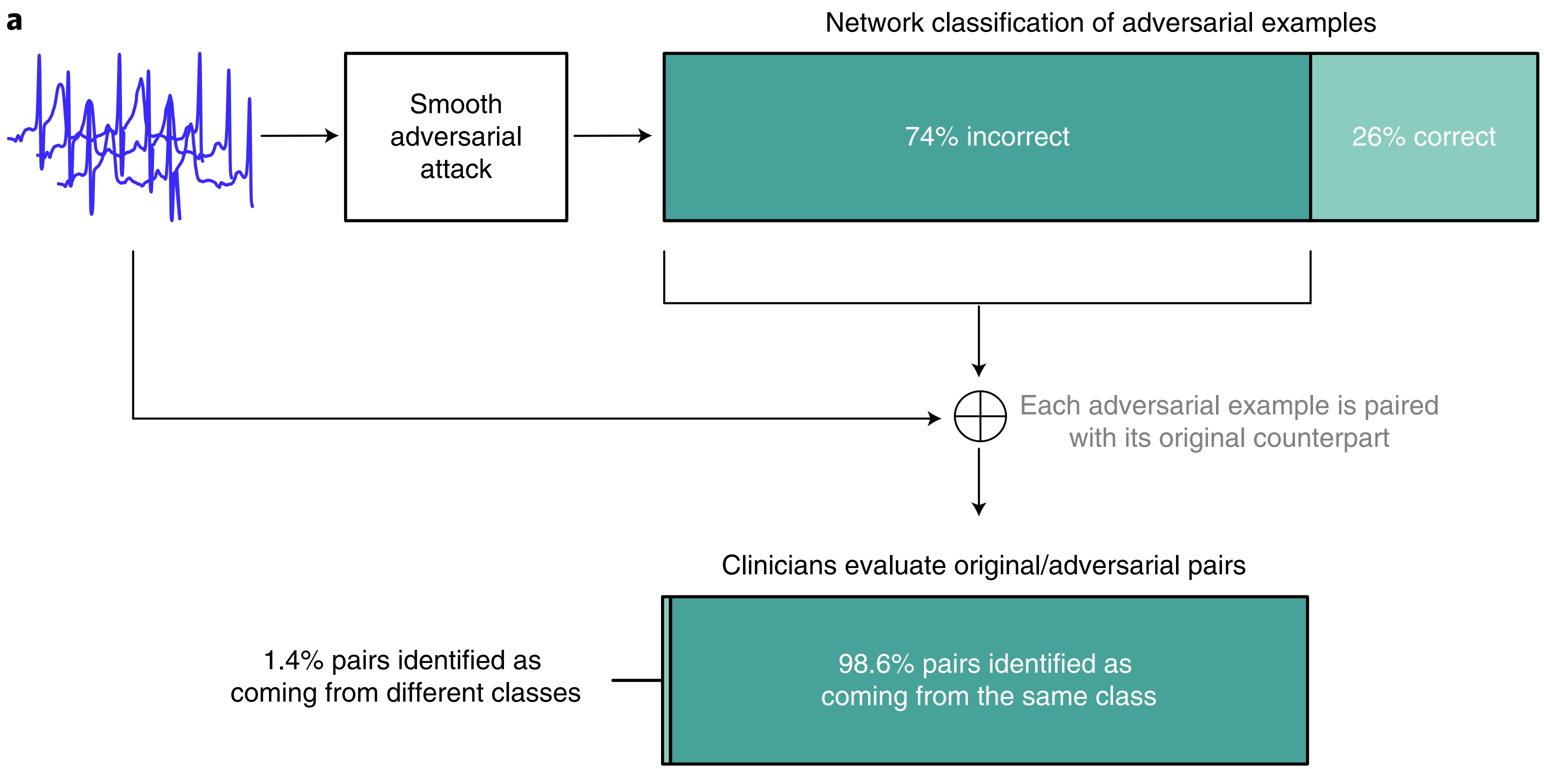
$$\theta'_i = \text{Clip}_{\mathbf{0},\epsilon}\left(\theta'_{i-1} + \alpha \text{sign}\left(\nabla_{\theta} L\left(f\left(\mathbf{x}_{\text{adv}}\left(\theta'_{i-1}\right), y\right)\right)\right)\right)$$

Gradient with respect to input

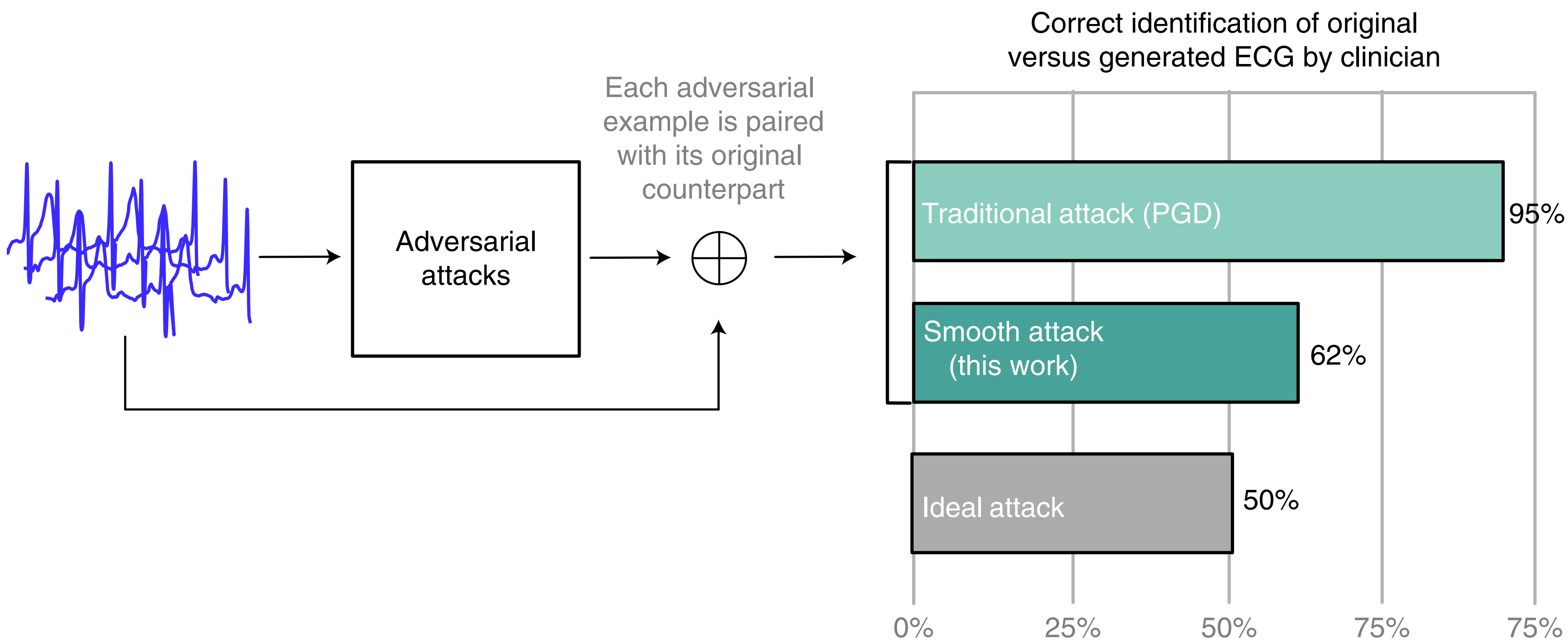
Adversarial examples for ECG data



How can we protect against an adversarial attack?



How good is the attack generation strategy?



How easy is it to change the class label?

Table 1 | Success rate of the targeted smooth attack method

		Target class			
		Normal (%)	**AF (%)**	**Other (%)**	**Noise (%)**
Original class	Normal	–	57	55	13
	AF	74	–	87	22
	Other	72	76	–	20
	Noise	79	64	57	–

The original class is the class into which the network classifies the signal before the adversarial attack. The target class is the class into which the adversarial attack aimed to make the network classify the signal after adding. The success rate is calculated as the percentage of examples from the original class that were misclassified by the network to the target class after the adversarial attack.

Summary

- Showcased that adversarial attacks are not restricted to images and can be adapted to clinical time-series data too
- Important to know this before making decisions from algorithms deployed in open-world scenarios:
  - Might be difficult to inject an adversarial attack into a radiologist’s software platform
  - Might be easy to inject an attack into a publicly visible and available platform

Questions?

- On Friday, Nikhil has kindly agreed to present on another technique for interpretability: LIME and Shapley values