Efficient Adversarial Sequence Generation for RNN with Symbolic Weighted Finite Automata

Mingjun Ma, Dehui Du, Yuanhao Liu, Yanyun Wang, Yiyang Li

Software Engineering Institute
East China Normal University, Shanghai, China

Presenter: Yuanhao Liu

01/03/2022, Virtual, SafeAI 2022-Workshop@AAAI-22

Outline

- Background
- Preliminaries
- Main approach
- Experiments
- Related work
- Conclusion and discussion

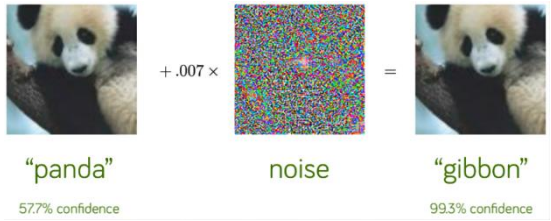
The Gist

- Efficient adversarial sequence generation approach for RNN by SWFA
  - Extract SWFA from RNN with the symbolic extraction algorithm Fast *k*-DCP
  - Perturb the symbolic input to generate adversarial sequences
- Adversarial sequences generated by our approach are more covert
  - Keep perturbation within the human-invisible range
- Implement adversarial sequence generation algorithm
  - Outperform the state-of-art attack methods with 112.92% improvement and 1.44 times speedup

Background

Robustness of DNN

Adversarial Attacks

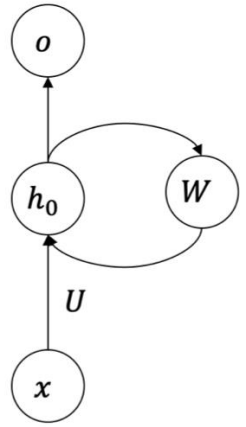


White-box attack
FGSM, PGD, etc.

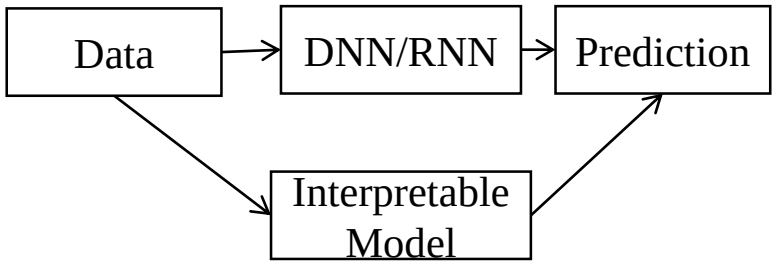
Black-box attack
NewtonFool, etc.

Not applicable for RNN.

The cyclic structure of **RNN** makes it difficult to craft adversarial examples on sequential data.



Interpretable Models



Derministic Finite Automata (DFA)

Probalistic Finite Automata (PFA)

Weighted Finite Automata (WFA)

Symbolic Weighted Finite Automata (SWFA)

Preliminaries

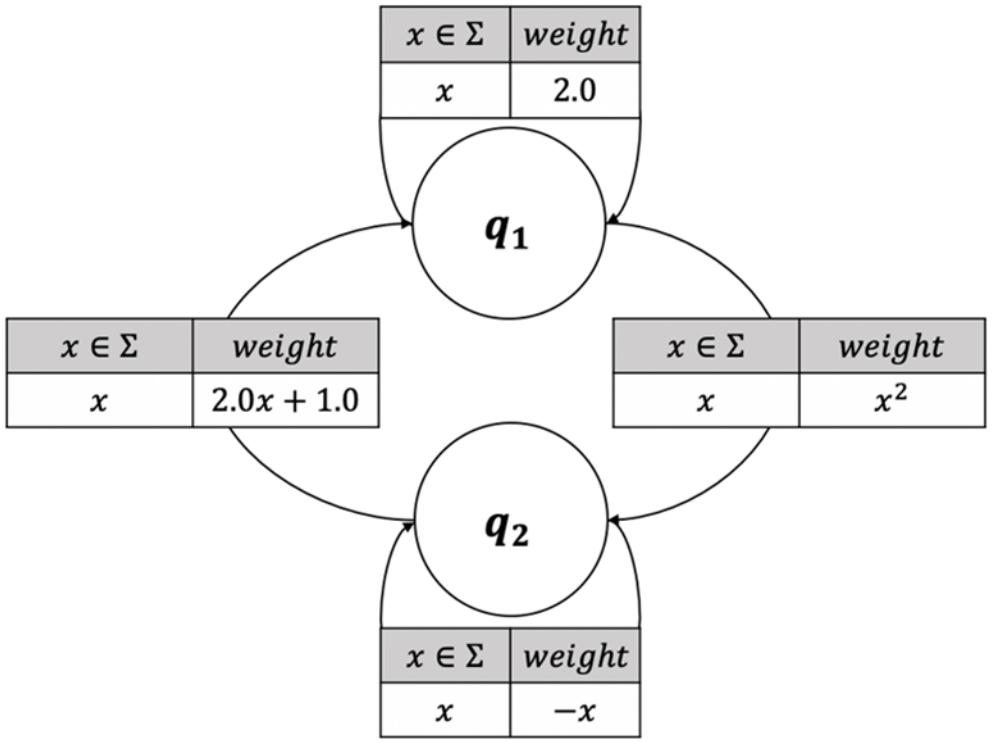
➢ Recurrent Neural Network

- RNN is denoted as a 6-tuple $R = (H, X, Y, h_0, f, g)$.

➢ Symbolic Weighted Finite Automata

- As well as WFA, SWFA can ***perform real-value operations***
- SWFA is denoted as a 5-tuple $\Upsilon = (G, Q, \alpha, \beta, A)$.

Symbolic Weighted Finite Automata

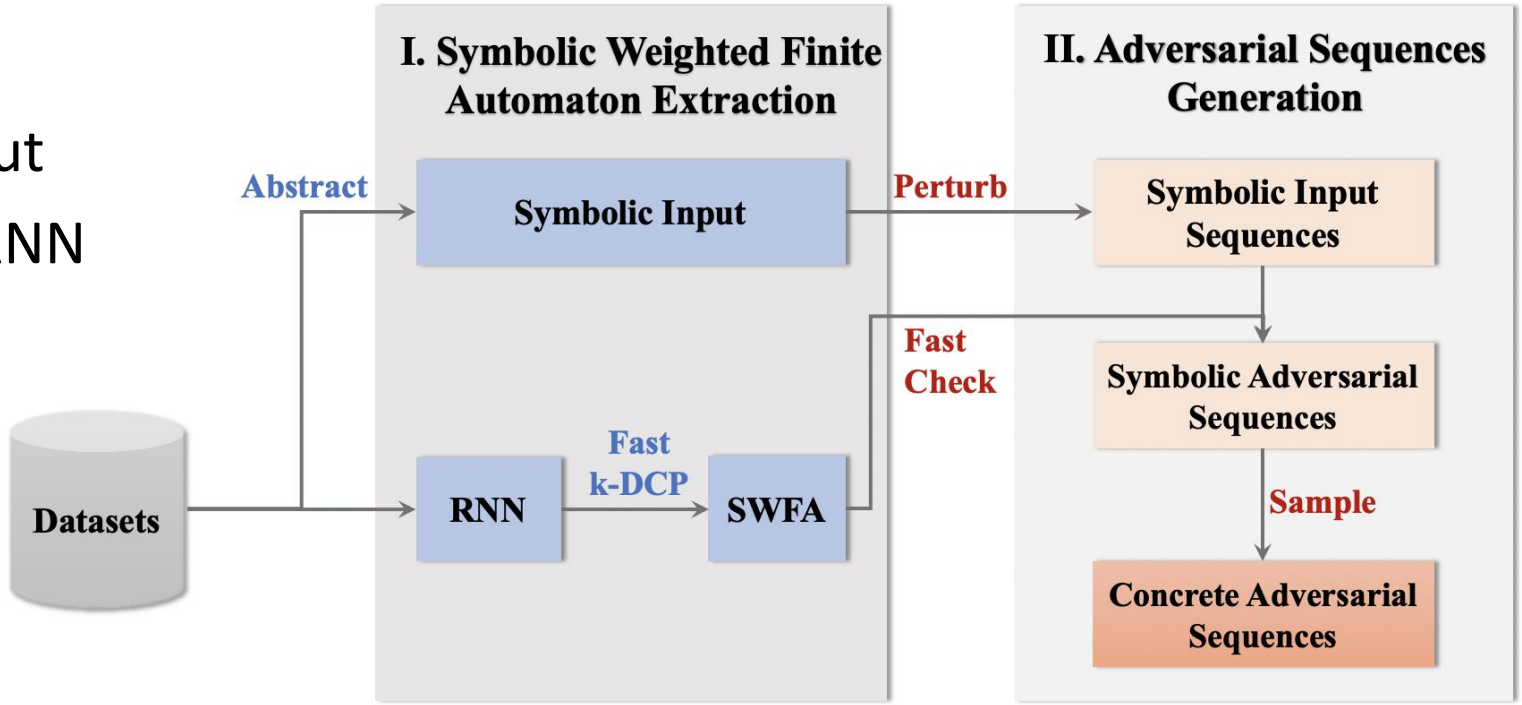


- Transition edges are labelled by ***functions***
- Enhance the ***abstraction ability*** of WFA
- Can deal with a ***possibly infinite alphabet*** efficiently
- ***First*** use SWFA for perturbation

Main Approach

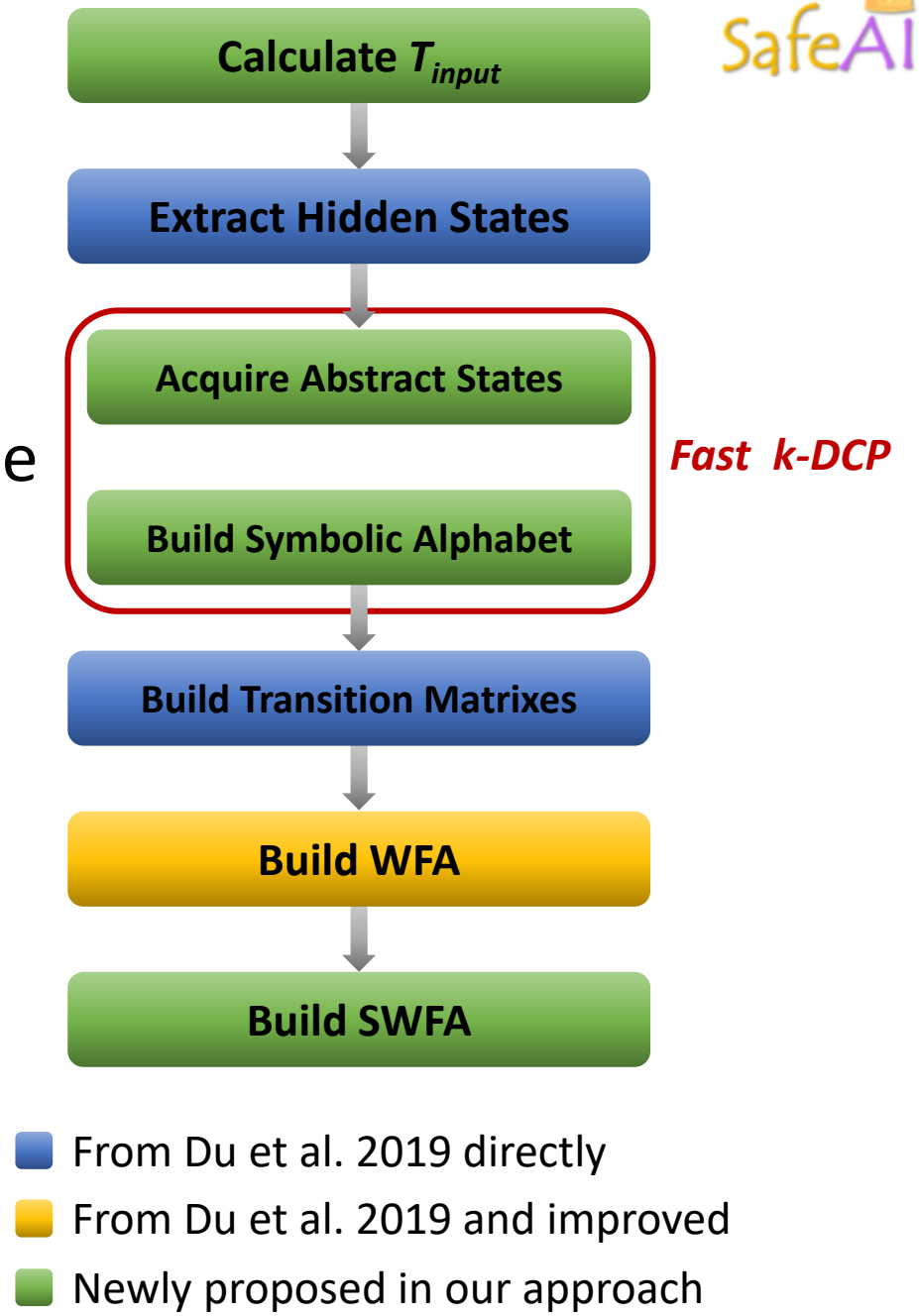
- Symbolic Weighted Finite Automata Extraction
  - Abstract the symbolic input
  - Abstract the SWFA from RNN

- Adversarial Sequences Generation by SWFA
  - Gain the symbolic input sequences
  - Screen out the symbolic adversarial sequences



Symbolic Weighted Finite Automata Extraction

- The *k-DCP* captures the top *k* ranked class labels as well as their prediction confidence levels.
- *High Efficiency*: Discarding the time-consuming k-means clustering and establishing symbolic blocks directly.
- *Symbolic Abstraction*: Extending to the infinite alphabet, which deals with input symbolically.



Fast *k*-DCP

(Our New Contribution)

- Time complexity: O(mns)
- Space complexity: $O(T^S)$
- Suitable for large-scale tasks

Algorithm 1: RNN-SWFA by Fast k-DCP

input : RNN $R = (H, X, Y, h_0, f, g)$
Input sequences W
K, T
output: SWFA $\Upsilon = (\mathcal{G}, Q, \alpha, \beta, A)$

Initialize $Q' = [s_0], \Sigma' = [\,], Q = [\,], \Sigma = [\,]$;
Initialize $A = [\,], \alpha = \pi q_0, \beta = [\,]$;
$d = ComputeDistance\,(W)$;
$T_{input} = \lceil 10/(d)/(|W| \times |w|) \rceil$;
for $w \in W$ **do**
  $s = [h^{(i)}(w)]_{i=0}^{|w|}$;
  for $i = 1$ **to** w **do**
    $Q'.add(s_i)$;
    $\Sigma'.add(w_{i-1})$
for $q' \in Q'$ **do**
  $Q.add(fkdcp^{K,T}(q'))$;
for $\sigma' \in \Sigma'$ **do**
  $\Sigma.add(fkdcp^{|\sigma'|,T_{input}}(\sigma'))$;
for $\sigma \in \Sigma$ **do**
  $A_\sigma = BuildTransitionMatrix(\sigma)$;
  $A.add(A_\sigma)$;
for $q \in Q$ **do**
  $\beta_q = 0$ with length $|L|$;
  for $q' \in Q'$ **do**
    if $fkdcp^{K,T}(q') == q$ **then**
      $\beta_q[argmax(g(q'))] += 1$;
  $\beta_q = \beta_q / \sum(\beta_q)$;
  $\beta.add(\beta_q)$;
$\mathcal{G} = GuardFuntionLearning(Q, \alpha, \beta, A)$;
return SWFA $\Upsilon = (\mathcal{G}, Q, \alpha, \beta, A)$

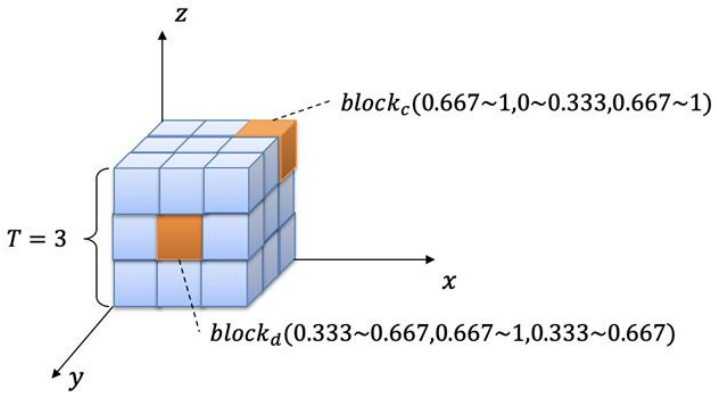
Du et al. 2019:

Zhang, X.; Du, X.; Xie, X.; Ma, L.; Liu, Y.; and Sun, M. 2021. Decision-Guided Weighted Automata Extraction from Recurrent Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11699–11707.

Adversarial Sequence Generation

(omitting details, cf. the paper)

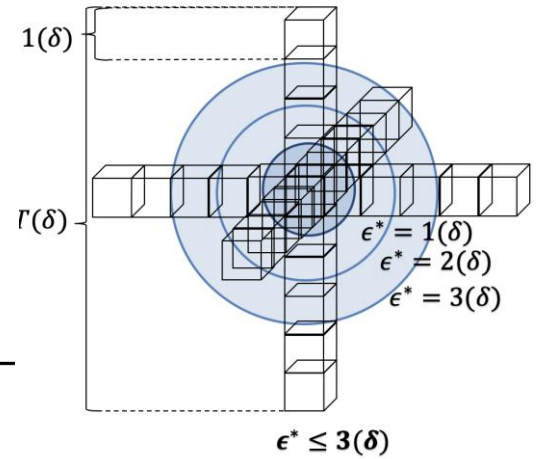
- Step 1: Set an appropriate T_{in}
- Step 2: Abstract input space i
  - Divided By Fast *k*-DCP
  - An interval [0,1] can be divide
- Step 3: Find Direction and Pe
- Step 4: Check ***"000-status"***
  - Represent the input exceeds t



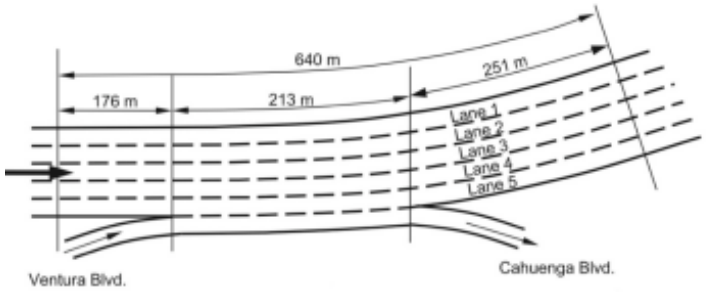
Algorithm 2:
Adversarial Sequence Generation by SWFA

input : Input x_0;
Continuous $c \in \{true, false\}$;
Number of nodes to be disturbed n;
Perturbation intensity $\epsilon^* \in \{1, 2, 3\}$.
output: Adversarial Sequence x'.

Initialize $x_0^{*'} = fkdcp(x_0)$; $x^{*'} = []$; $\delta = 0$;
while $\delta < \epsilon^*$ **do**
  $Nodes = NodesSearch(c, n)$;
  for $node \in Nodes$ **do**
    $dir = FindDirectionbyImportance(x_0^{*'})$;
    $x_{pert} = Pert(x_0^{*'}, node, dir, \delta)$;
    if $verifyBySWFA(x_{pert})$ **then**
      $x^{*'}.add(x_{pert})$;
  $\delta = \delta + 1$;
$x' = Sampling(x^{*'})$;
return x'



Experiment Setting



Public Datasets:

- NGSIM
  - Next Generation Simulation (NGSIM) program collected detailed vehicle trajectory data on southbound US 101 through a network of synchronized digital video cameras.
- UCR time-series datasets
  - Introduced in 2002, open source time-series data, with at least one thousand making use of these datasets.

Experiment Setting

Our Datasets:

- *ADD (Proposed by this paper)*
  - Autonomous Driving Datasets Generated by Carla (Zhang et al. 2021)

Car is turning left

Serial number	Time step	Longitude Coordinate	Latitude Coordinate	Vehicle Width(m)	Head turn
1	1	-41.2	50.2	2	Left
	2	-43.1	50.5	2	
	...	-44.5	50.6	2	
	40	-48.6	50.4	2	
2	1	-48.6	50.4	2.5	Right
	2	-44.5	50.1	2.5	
	...	-43.1	49.2	2.5	
	40	-41.2	49.8	2.5	
...					
3000	1	100.2	1.2	2.5	Straight
	2	101.9	2.3	2.5	
	...	100.1	2.1	2.5	
	40	99.5	2.0	2.5	

Our data structure

Experiment Ⅰ: *RNN-SWFA Extraction*

Table 1: Comparison between SWFAs extracted by Fast k-DCP on various time-series data

Datasets	AoR(%)	AoS(%)	ET(s)	RT(s)	ST(s)
ADD	99/97	**89/79**	421.536	4.982	3.214
NGSIM	91/86	**77/73**	28.704	3.016	2.971
PPOAG	75/88	35/43	2.667	0.224	0.443
CT	53/74	53/73	0.026	0.005	0.004
EQ	82/75	**82/76**	7.229	1.112	1.194

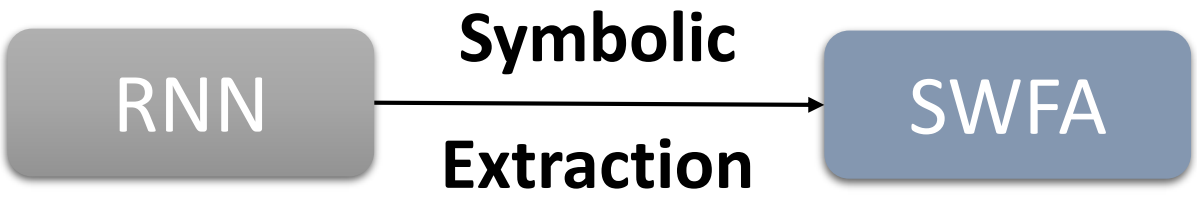
AoR: Accuracy of RNN (training/test)

AoS: Accuracy of SWFA (training/test)

ET: Extraction Time of SWFA

RT: Running Time of RNN

ST: Running Time of SWFA



- $RNN_{Acc} \approx SWFA_{Acc}$
- $RNN_{RunningTime} \approx SWFA_{RunningTime}$

- Reuse the time-consuming extraction
- Work in **infinite alphabets**

Experiment Ⅱ : *SWFA-based adversarial sequence generation*

Table 2: Comparison between abstraction-based adversarial sequence generation approach and other adversarial attacking algorithms on the autonomous driving dataset

Category	White Box						Black Box			Our Approach		
Methods	FGSM			PGD			NewtonFool			AbASG		
Perturbation(δ)	1	5	10	1	5	10	1	2	3	1	2	3
ASR(%)	0.00	0.33	21.66	0.00	0.33	3.8	11.33	17.66	25.23	**20.52**	**34.36**	**53.72**
Time(s)	-	3.15	10.00	-	10.68	22.58	42.25	26.85	26.7	**39.94**	**20.42**	**18.55**

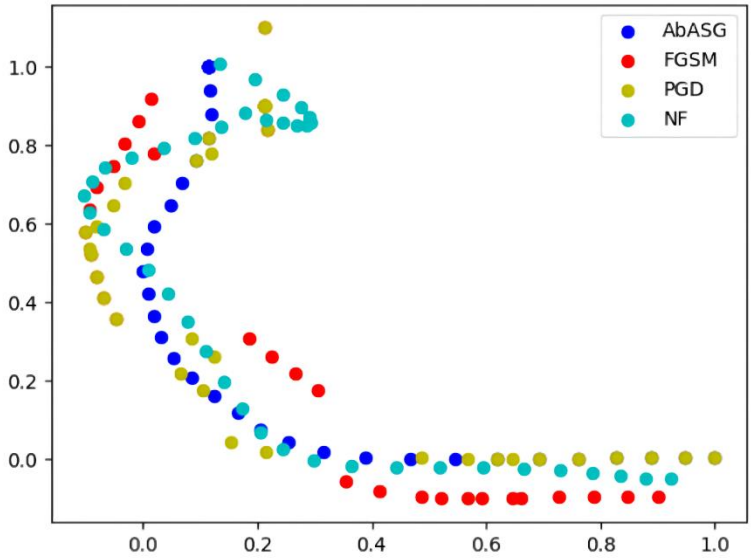
Our Approach

81.11%, 94.56% and 112.92%

Outstanding success rate

1.44 times

Fast Speed



Related Work

- *More efficient in generating adversarial sequences*
- *With more subtle perturbations*

- *Take advantage of the real-value operation ability of WFA to simulate RNN.*
- *Use the symbolic characteristics of SWFA, which enhances generalization.*

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. *arXiv 1412.6572*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jang, U.; Wu, X.; and Jha, S. 2017. Objective Metrics and Gradient Descent Algorithms for Adversarial Examples in Machine Learning. In *Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, December 4-8, 2017*, 262–277. ACM.

Ayache, S.; Eyraud, R.; and Goudian, N. 2018. Explaining Black Boxes on Sequential Data using Weighted Automata. In Unold, O.; Dyrka, W.; and Wieczorek, W., eds., *Proceedings of the 14th International Conference on Grammatical Inference, ICGI 2018, Wrocław, Poland, September 5-7, 2018*, volume 93 of *Proceedings of Machine Learning Research*, 81–103. PMLR.

Conclusion

Main Contribution:

- The novel ***Fast k-DCP*** symbolic extraction algorithm
- ***Efficient adversarial sequence generation*** by SWFA

Main Advantage:

- Applicable to generate ***covert*** adversarial sequences
- Perturbation within ***human-invisible range***
- Suitable for ***Spatio-temporal*** sequential tasks

Discussion

Drawbacks:

- Not yet adapting to large-class sequential data
- Should study on various datasets.

Future work:

- Further optimize our approach.
- Investigate the reachability analysis of SWFA.
- Explore more valuable properties of SWFA for improving efficiency.

Thank you for your attention

➢Questions?(dhdu@sei.ecnu.edu.cn)

- Acknowledgements: Financial support for this work, provided by the National Natural Science Foundation of China under Grant No.61972153, the Key projects of the Ministry of Science and Technology under No. 2020AAA0107800, is gratefully acknowledged.