Attacking and Defending Machine Learning based Intrusion Detection Systems

Student:
Jehoshua Hanky Pratama, Didik Sudyana

Advisor:
Professor Ying-Dar Lin

High Speed Network Lab
National Yang Ming Chiao-Tung University, Taiwan

Outline

- **Motivation**
- **Background**
- **Issues**
  - Inter-technique transferability
  - Single vs. ensemble
  - Adversarial training
- **Notations**
- **Problem Statements**
  - Single and ensemble ML-based IDS
  - Adversarial attack dataset generation and selection
  - Adversarial trained ensemble ML-based IDS
  - Best approach
- **Related Works**
  - Defense comparison
  - Attack applicability to IDS
- **Solutions**
  - Solution overview
  - F1 score for basic model
  - Double fault and Kappa statistics filter for ensemble team
  - Exhaustive comparison
  - F1 score for adversarial model threshold
  - Double fault and Kappa statistics filter for ensemble adversarial team
  - Best approach selection
- **Evaluation**
  - Testbed configuration
  - Transferability property
  - Adversarial defense
- **Results**
- **References**

Motivation

- Adversarial Attack
  - Can fool machine learning models [1]
- Adversarial Attack on IDS
  - Affect ML-based IDS
  - "Double attack"
    - Fool machine learning based IDS, then attack the network
- Adversarial Defense
  - Mostly defense techniques for image classification
  - Existing defense techniques focus on the same model attack
  - Attack transferability property has been discovered

Background - Network Security

- Has become an important issue for everyone’s life [2]

- Intrusion Detection System (IDS):
  - Traditional IDS
    - Signature-based
    - Anomaly-based
  - ML-based IDS
    - Has a satisfactory detection level
    - Detect more attack variants

Background - Adversarial Machine Learning

- Machine learning can be exploited by adversarial attack
- Example of adversarial attack :
  - Input an adversary data to a classifier
    - Causing misclassification
- Degrade the machine learning performance
- Extensively explored in image classification and spam detection
  - Less in intrusion detection [3]

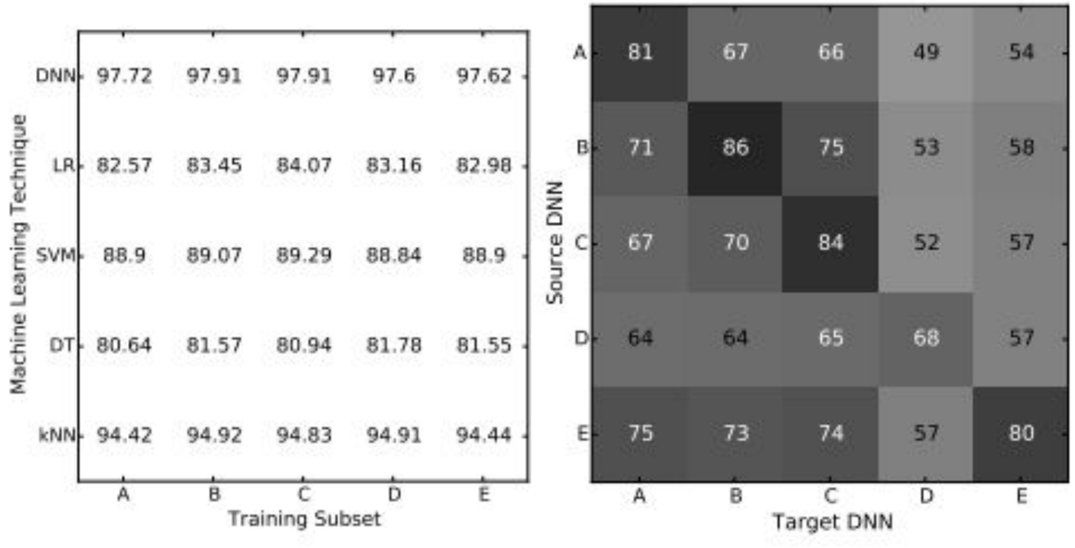
Background - Adversarial Attack

- **Poisoning attack**
  - Manipulating training data [4]
    - Injecting adversarial points into the training set

- **Evasion/input attack**
  - Manipulates test samples to have them misclassified [5]

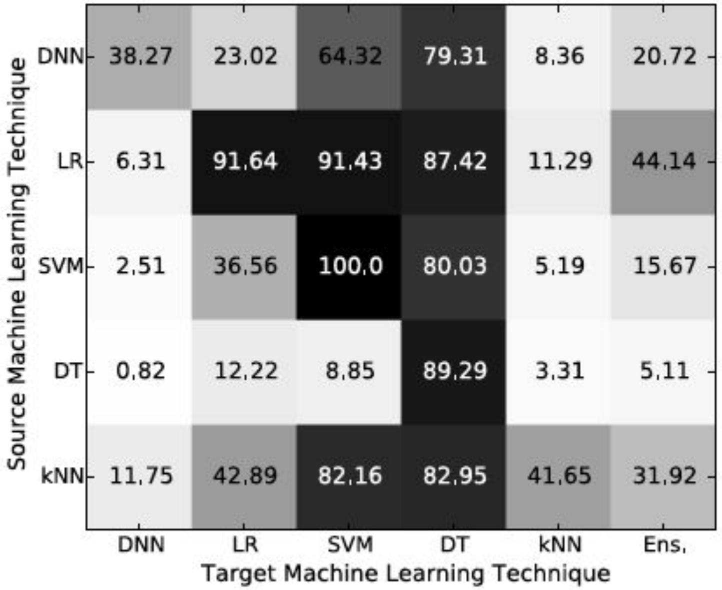
Background - Adversarial Attack Characteristics

- **Attack transferability:**
  - Adversarial data can be used to fool more than one model [6].
  - If it succeeds to fool a specific model, it can succeed to fool another model trained by the same dataset



(a) Model Accuracies (b) DNN models

Intra-technique Transferability [6]



Inter-technique Transferability [6]

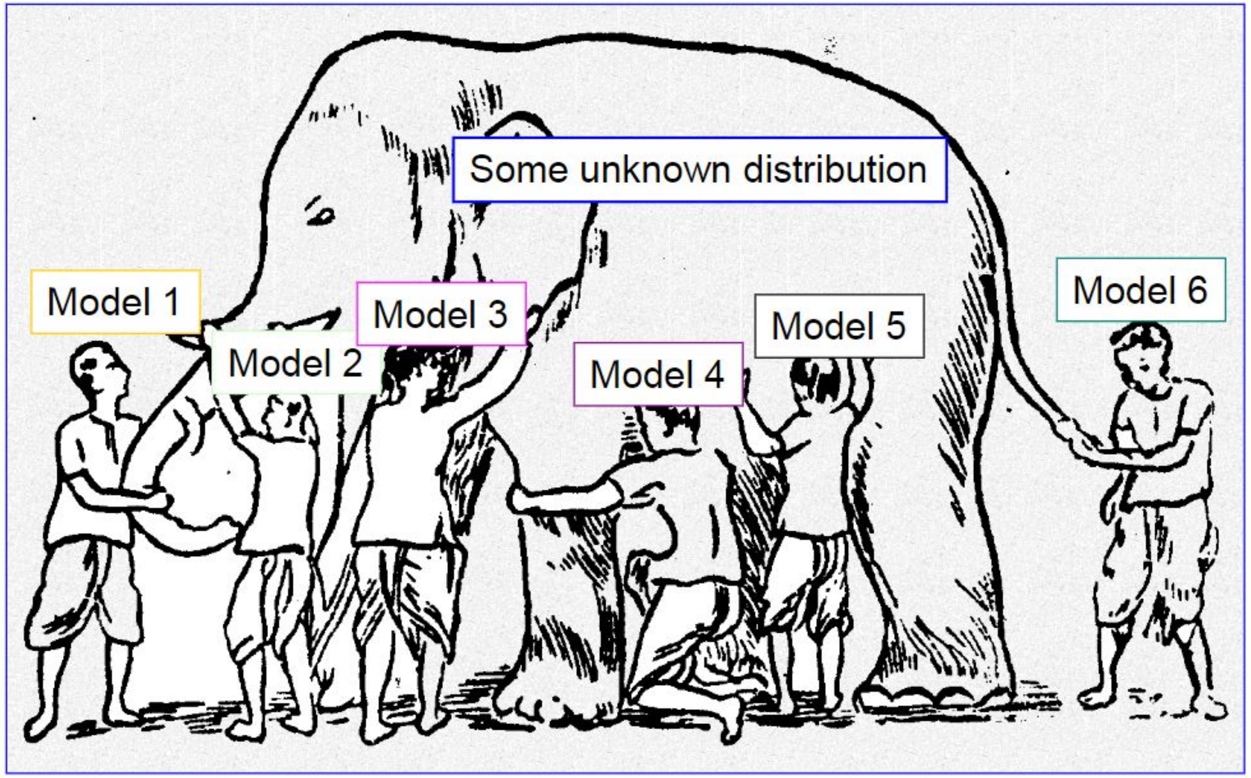
Background - Adversarial Defense

- **Adversarial training**
  - Include the adversarial data to the training [7]

- **Ensemble learning**
  - Combination of models to make the system robust [8]

Background - Ensemble Learning

- Ensemble learning:



Ensemble gives the global picture!

Gao, J., et all., (2010)

Background - Diversity

- The key of a powerful ensemble: Model diversity [9]

- The diversity can help each procedure to guarantee a totally good ML [8]
    - Diversity in training
    - Diversity in model
    - Diversity in decision

Background - Diversity

- Diversity in training
    - It provides more information for the model [10]

- Diversity in model
    - It makes each model capture unique or complement information [10]

- Diversity in decision
    - It provides multiple choices each of which corresponds to a specific plausible local optimal result [10]

Background - Measurement Score (1/2)

Kappa Statistics

- Remove bad ensemble teams with high Kappa values [11]
  - Indicating low level of disagreement diversity

- The example of Kappa agreement score [11]:
  - Poor agreement : < 0.20
  - Fair agreement : 0.20 to 0.40
  - Moderate agreement : 0.40 to 0.60
  - Good agreement : 0.60 to 0.80
  - Very good agreement : 0.80 to 1.00

Background - Measurement Score (2/2)

Double-Fault Measurement

- Probability that both classifiers make the same wrong prediction [12]
- Remove bad ensemble teams with high double-fault values [12]
    - A lower value means the classifiers are less likely to make the same error

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}$$

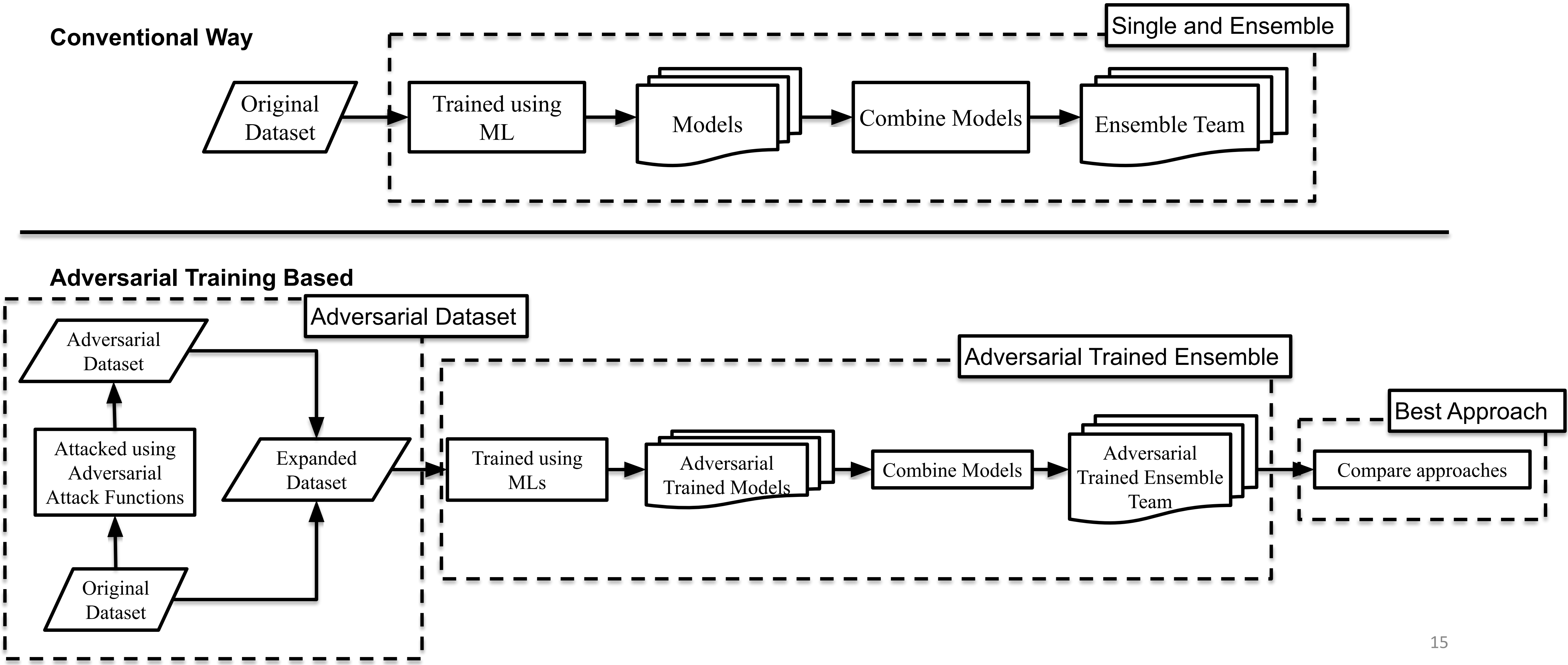
	C_k correct	C_k wrong
C_i correct	N^{11}	N^{10}
C_i wrong	N^{01}	N^{00}

Issues - Adversarial Defense for ML-based IDS

- Inter-technique transferability
  - Transfer adversarial attack function to another model
- Single vs. ensemble
- Adversar

Approach	Single vs. Ensemble	Adversarial Training
Basic	Single	No
Ensemble	Ensemble	No
Adversarial	Single	Yes
Ensembled Adversarial	Ensemble	Yes

Problems – Overview



Problem Statements - Single and Ensemble

- Input:
  - -An IDS training dataset which consists of a set of labeled input data
  - -Machine learning algorithms
  - -A testing dataset

- Output:
  - -Decide the best single model and the best ensemble team

- Objective:
  - -Highest F1 score on the model tested using testing dataset

- Constraint:
  - - None

Problem Statements – Adversarial Dataset

Generation and Selection

- Input:
  - -An IDS dataset which consists of a set of labeled input data
  - -Adversarial attack functions
  - -All single ML-based models

- Output:
  - -Choose functions to generate expanded dataset

- Objective:
  - -Lowest average F1 score when models tested on adversarial attacked dataset

- Constraint:
  - - None

Problem Statements – Adversarial Trained Ensemble

- Input:
  - Expanded training dataset which consist of a set of clean input data and adversarial attacked input data with their own labels
  - Expanded testing dataset
  - Machine learning algorithm
  - Single ML-based models
  - Ensemble Team

- Output:
  - Decide the best adversarial trained single model and the best adversarial trained ensemble team

- Objective:
  - Maximize the difference of summed F1 scores between single models and ensemble models tested in both clean and adversarial attacked dataset

- Constraint:
  - None

Problem Statements – Best Approach

- Input:
  - -Best models from all 4 approaches: Single, ensemble, adversarial, ensemble adversarial.
  - -Expanded testing dataset

- Output:
  - -Decide the best approach to defend IDS against adversarial attack

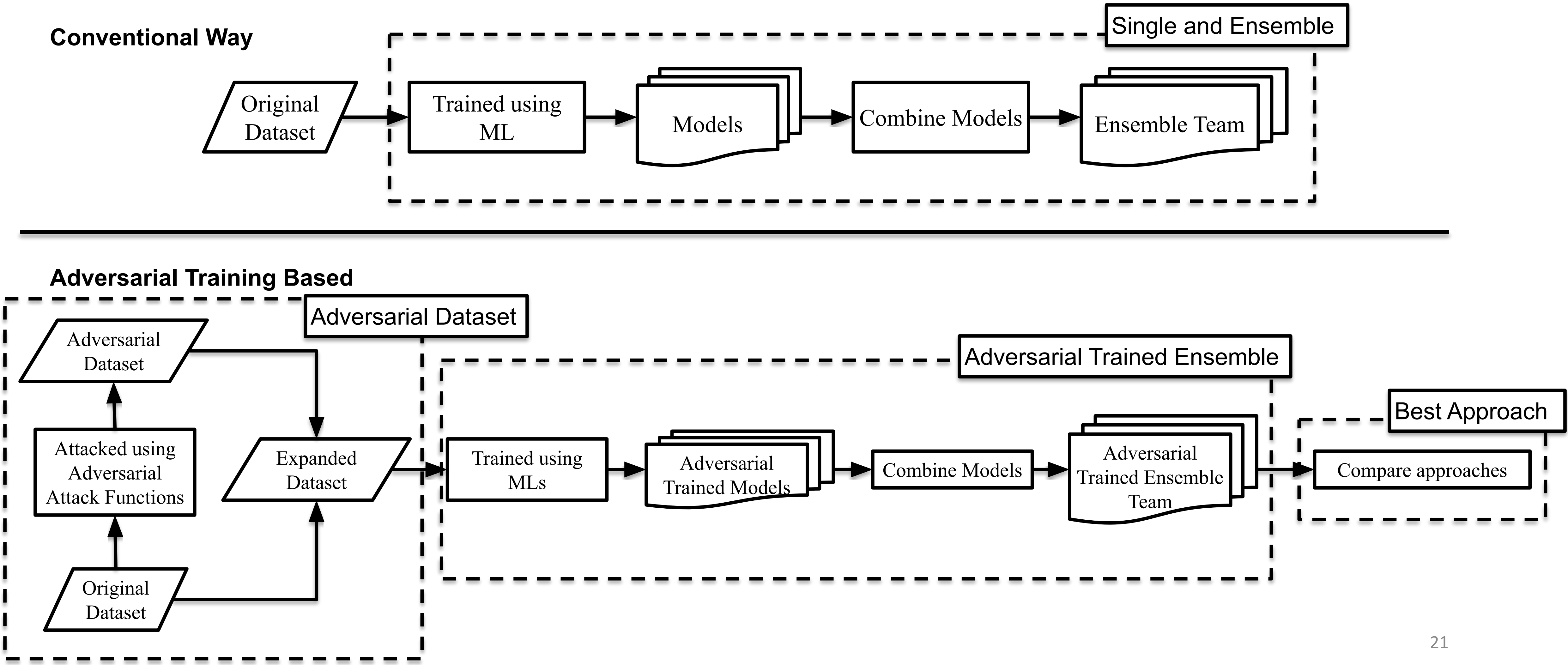
- Objective:
  - -Minimize the degradation of F1 score when tested using the expanded testing dataset

- Constraint:
  - - None

Notations

Category	Name	Notation	Note
Dataset	Dataset	D	$D = \{(x_i, y_i), i = 1, 2, 3, \ldots, n\}$; $D = D^R \cup D^T$; $R \cup T = \{1, 2, 3, \ldots, n\}$
	Dataset for Testing	D^T	
	Dataset for Training	D^R	
	Expanded Dataset with Adversarial Samples	D^E	$D^E = D \cup D^+$
	Expanded Dataset for Testing	D^{ET}	
	Expanded Dataset for Training	D^{ER}	
	Data Input	x_i	
	Label	y_i	
Machine Learning	Number of ML Algorithm	N_{ML}	
	ML Algorithm	ML_j	
	ML Model		
	Best ML Model		Model with the highest F1 score
	ML Model with Adversarial Training		
	Best ML Model with Adversarial Training		Adversarial Trained Model with the highest F1 score
	Ensemble Team		
	Best Ensemble Team		Ensemble Team with the highest F1 score
	Ensemble Team with Adversarial Training		
	Best Ensemble Team with Adversarial Training		Adversarial Trained Ensemble Team with the highest F1 score
	Best Approach		Approach with the lowest F1 score difference
Attack	Adversarial Attack Dataset	D^+	
	Adversarial Attack Data		
	Number of Attack Technique	N_F	

Problems – Overview



Problem Statements - Single and Ensemble

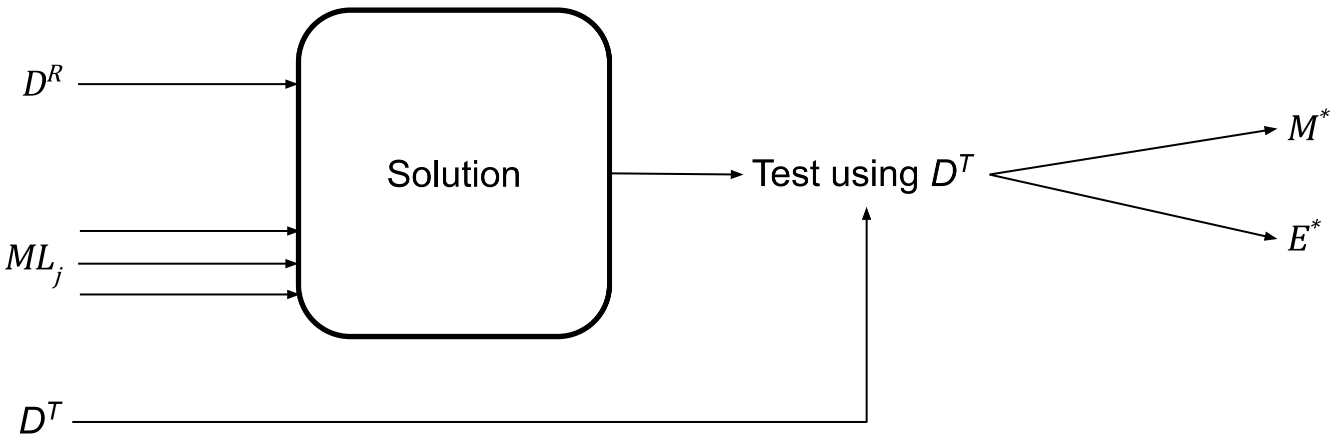
- Input:
  - An IDS dataset training D^R which consists of a set of x_i with y_i
  - Machine Learning Algorithm ML_j
  - A testing dataset D^T

- Output:
  - Decide the best single model M^* and the best ensemble team E^*

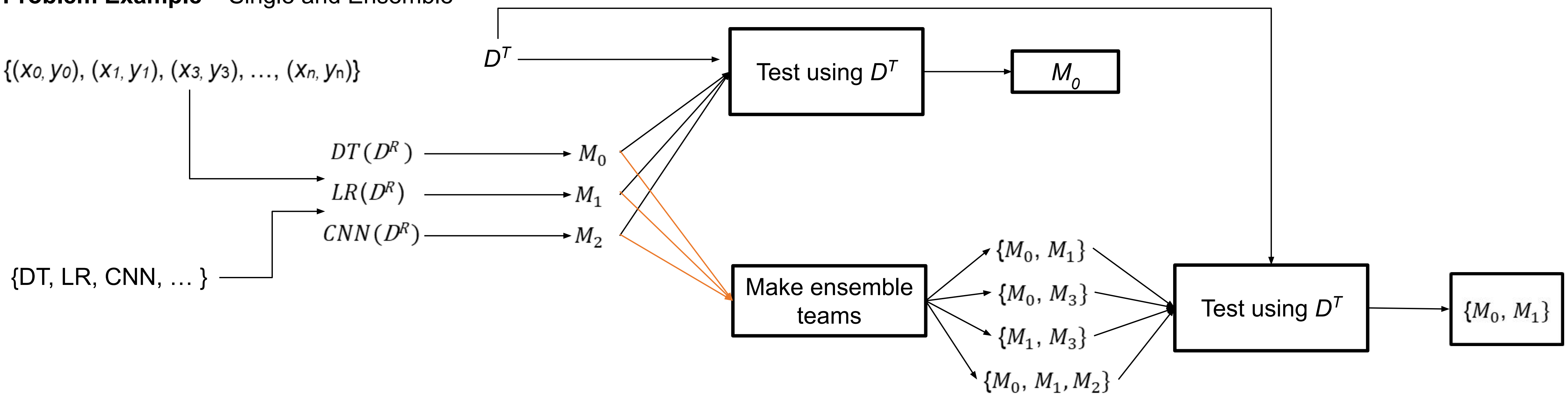
- Objective:
  - Highest F1 score on the model tested using D^T

- Constraint:
  -

Problem Figure – Single and Ensemble



Problem Example – Single and Ensemble



Problem Statements – Adversarial Dataset

Generation and Selection

- Input:
  - An IDS dataset D which consists of a set of x_i with y_i
  - Adversarial Attack Functions F
  - All single ML-based models M

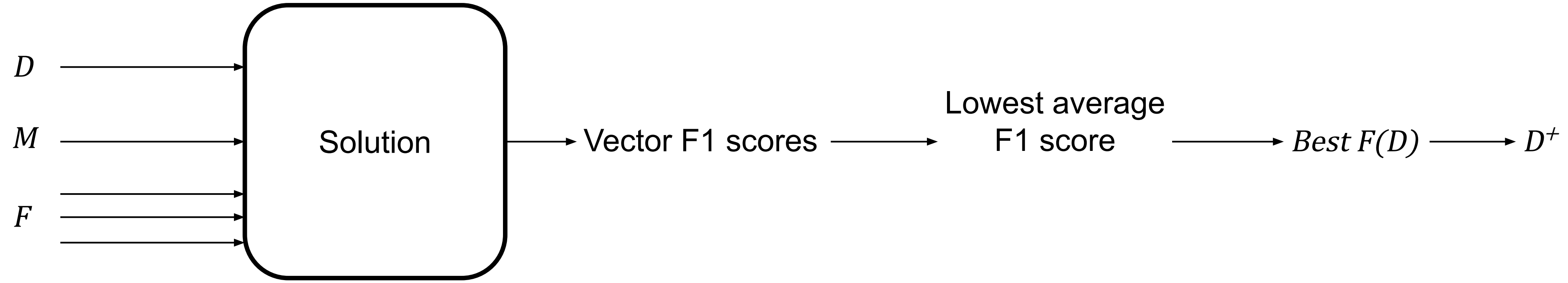
- Output:
  - Choose function from F to generate expanded dataset D^+

- Objective:
  - Lowest average F1 scores when M tested on D^+

- Constraint:
  -

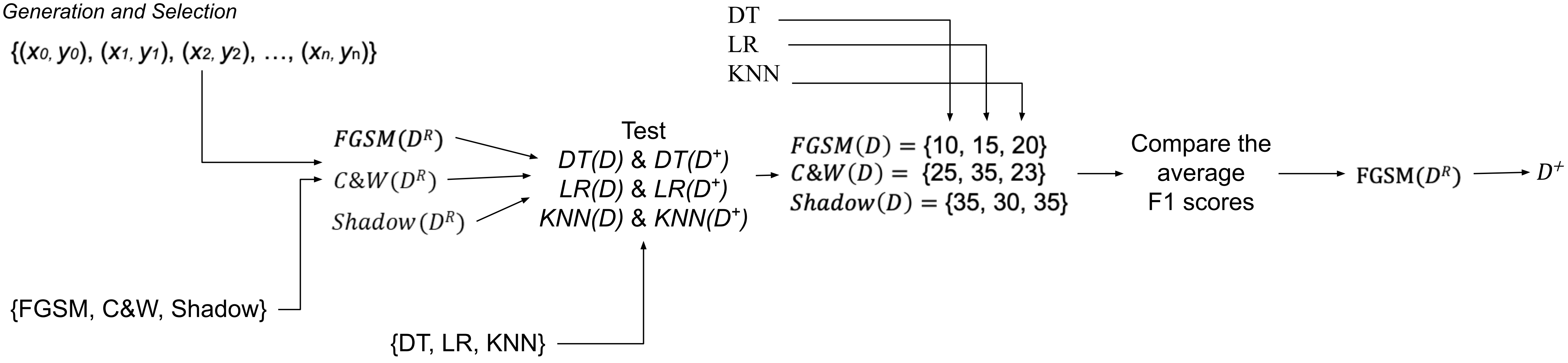
Problem Figure – Adversarial Attack Dataset

Generation and Selection



Problem Example – Adversarial Attack Dataset

Generation and Selection



Problem Statements – Adversarial Trained Ensemble

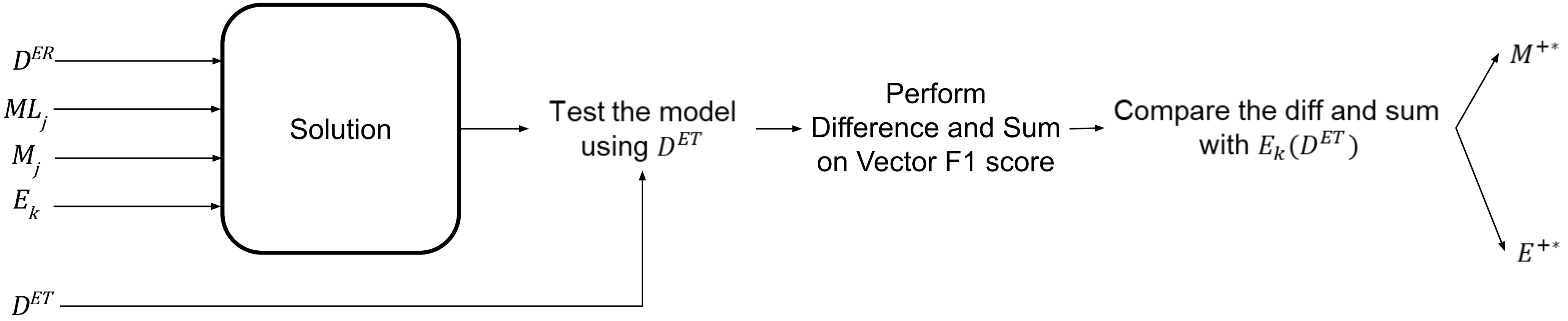
- Input:
  - Expanded training dataset D^{ER} which consist of x_i and x_i^+ with label y_i
  - Expanded testing dataset D^{ET}
  - Machine Learning Algorithm ML_j
  - Single ML-based models M_j
  - Ensemble Team E_k

- Output:
  - Decide the best adversarial train single model $M^{+*}$and the best adversarial train ensemble team E^{+*}

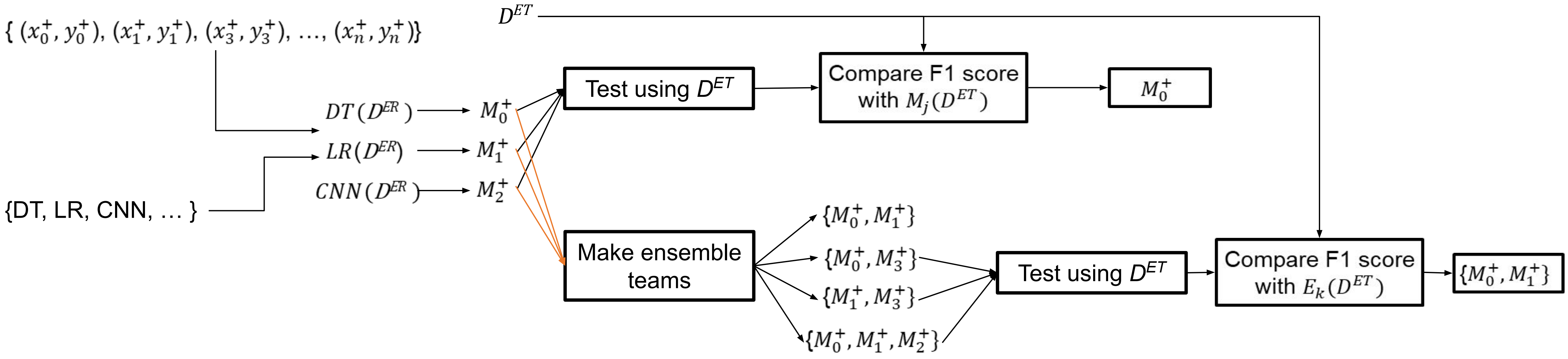
- Objective:
  - Maximize the difference of summed F1 scores between $M_j^+(D^{ET})$ and $M_j(D^{ET})$ also between $E_k^+(D^{ET})$ and $E_k(D^{ET})$

- Constraint:
  -

Problem Figure – Adversarial Trained Ensemble



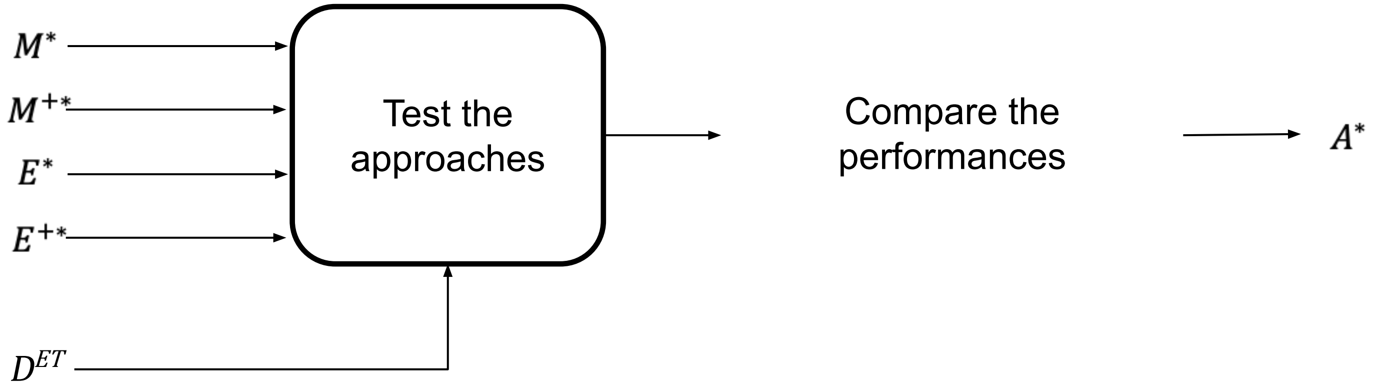
Problem Example – Adversarial Trained Ensemble



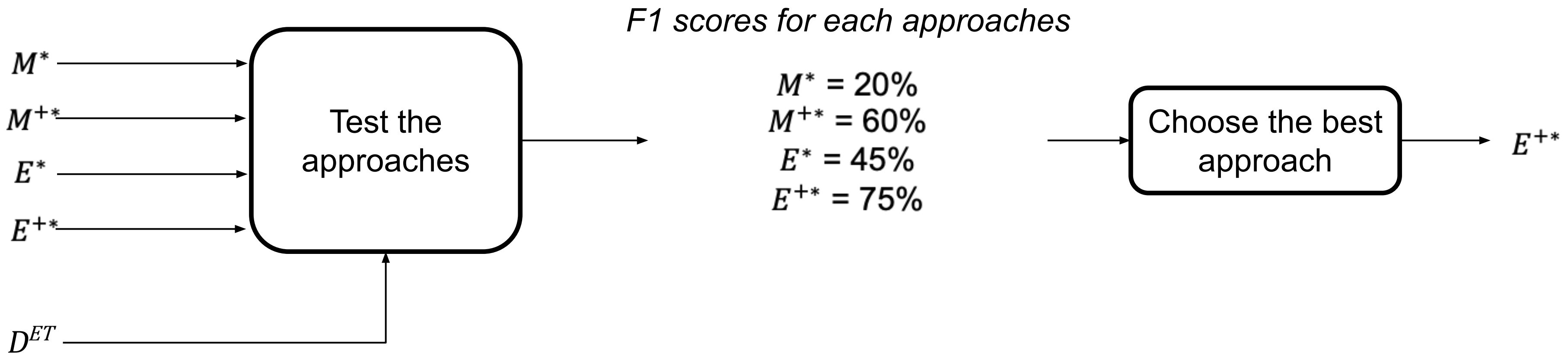
Problem Statements – Best Approach

- Input:
  - Best models from each approaches. M^*, M^{+*}, E^*, E^{+*}.
  - Expanded testing dataset D^{ET}
- Output:
  - Decide A^*
- Objective:
  - Minimize the degradation of F1 score when tested using D^{ET}
- Constraint:
  -

Problem Figure – Best Approach



Problem Example – Best Approach



Related Works – Comparison Defense

Paper	Adversarial Training	Ensemble Learning	Attack Techniques	Classifiers	Diversity Area			Measuring Diversity Model	Transferability Analysis
					Training	Model	Decision		
[13]	-	-	FGSM, BIM, PGD	FNN and SNN	V	-	-	-	-
[14]	-	-	FGSM, BIM, C&W, PGD	Random Forest and Nearest Neighbor	V		-	-	
[15]	V	-	C&W, FGSM, BIM, PGD, Deepfool	ANN and Random Forest	V	-	-	-	-
[16]	V	-	JSMA	Random Forest and J48	V	-	-	-	-
[17]	-	V	Alter some features	Random Forest	-	V	-	-	-
[18]	-	V	FGSM, JSMA, C&W, Deepfool, BIM and PGD	SVM, Decision Tree, DNN with voting	-	V	V	-	-
Ours	V	V	Decision Tree Attack, BIM, JSMA, Deepfool, FGSM, PGD, C&W, Zoo Attack	Decision Tree, SVM, KNN, XGBoost, LR, DNN, Keras	V	V	V	Kappa & Double-Fault	V

Related Works –Attack Applicability to IDS (1/2)

Paper	Attack Technique	Domain	IDS Compatibility
[19]	Shadow Attack	Image	-
[20]	Wasserstein Attack	Image	-
[21]	Brendel & Bethge Attack	Image	-
[22]	Square Attack	Image	-
[23]	Threshold Attack	Image	-
[6]	Decision Tree Attack	Image	[6, 35]
[24]	Basic Iterative Method	Image	[13]
[25]	Jacobian Saliency Map	Image	[16, 29, 30, 31, 1]
[26]	Deep Fool	Image	[1]
[5]	Fast Gradient Method	Image	[30, 31, 1, 32, 13, 34]
[27]	Projected Gradient Descent	Image	[13, 34]
[27]	Carlini & Wagner	Image	[31, 1, 33]
[28]	Zoo Attack	Image	[33]

Key Idea from this result:

1. There are 8 attack techniques applicable to IDS.
2. Papers listed on the IDS Compatibility column are the ones that already proved those attacks are applicable.

Related Works –Attack Applicability to IDS (2/2)

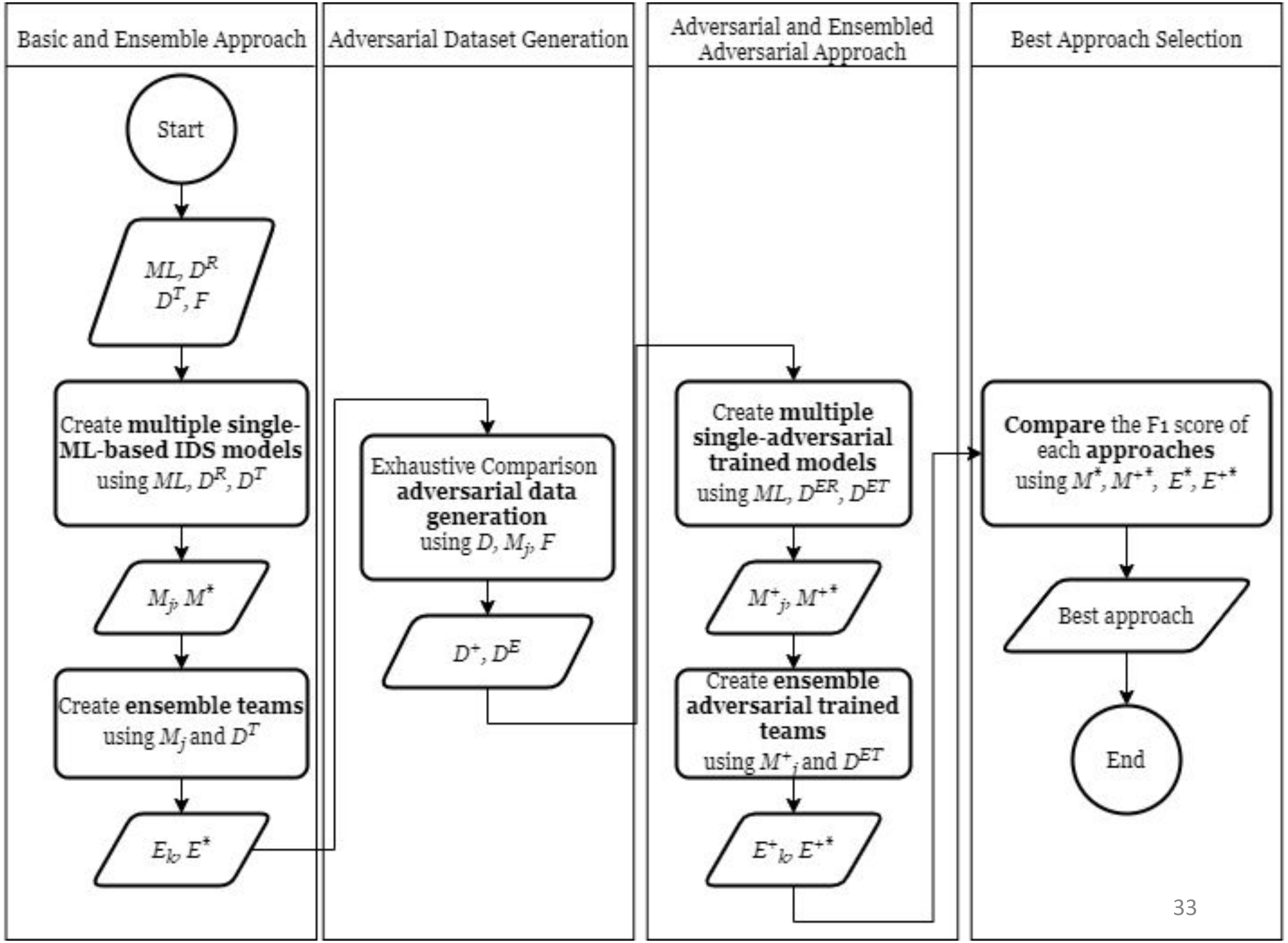
Attack Technique	Decision Tree	KNN	LR	SVM	XGBoost	DNN	Keras
Shadow Attack							
Wasserstein Attack							
Brendel & Bethge Attack							
Square Attack							
Threshold Attack							
Decision Tree Attack	ART	-	-	-	-	-	-
Basic Iterative Method			ART				
Jacobian Saliency Map							DeepIDS / Rambasnet
Deep Fool							DeepIDS / Rambasnet
Fast Gradient Method							DeepIDS / Rambasnet
Projected Gradient Descent			ART	ART			
Carlini & Wagner			ART	ART			
Zoo Attack	ART			ART	ART		

**ART = Adversarial Robustness Toolbox*

Overview Solution

There are 4 **sections** in this solution:

- Basic and Ensemble Approach
- Adversarial Dataset Generation
- Adversarial and Ensembled Adversarial Approach
- Best Approach Selection

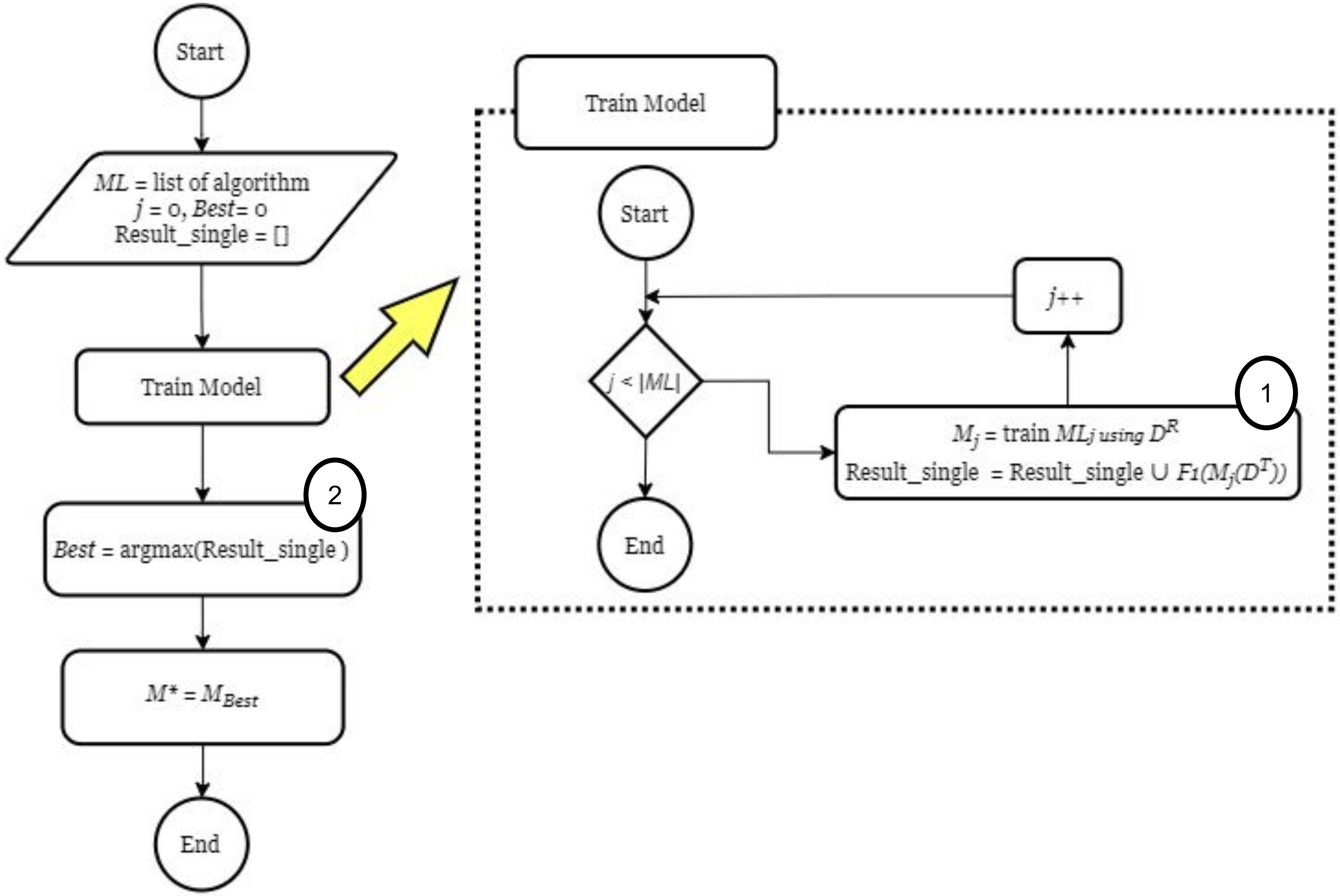


Problem: Basic Ensemble Approach Model Creation

Solutions – F1 Score for Basic Model

There is 1 loop in this solution:

- Loop by the number of machine learning algorithms



Example runs

ML = {DT, LR}
Result_single = [ ]

1 M_0 = DT({(x_0, y_0), (x_1, y_1), (x_3, y_3), …, (x_n, y_n)})
Test M_0
Append the result to Result_single
Result_single = [0.99]

1 M_1 = LR({(x_0, y_0), (x_1, y_1), (x_3, y_3), …, (x_n, y_n)})
Test M_1
Append the result to Result_single
Result_single = [0.99, 0.97]

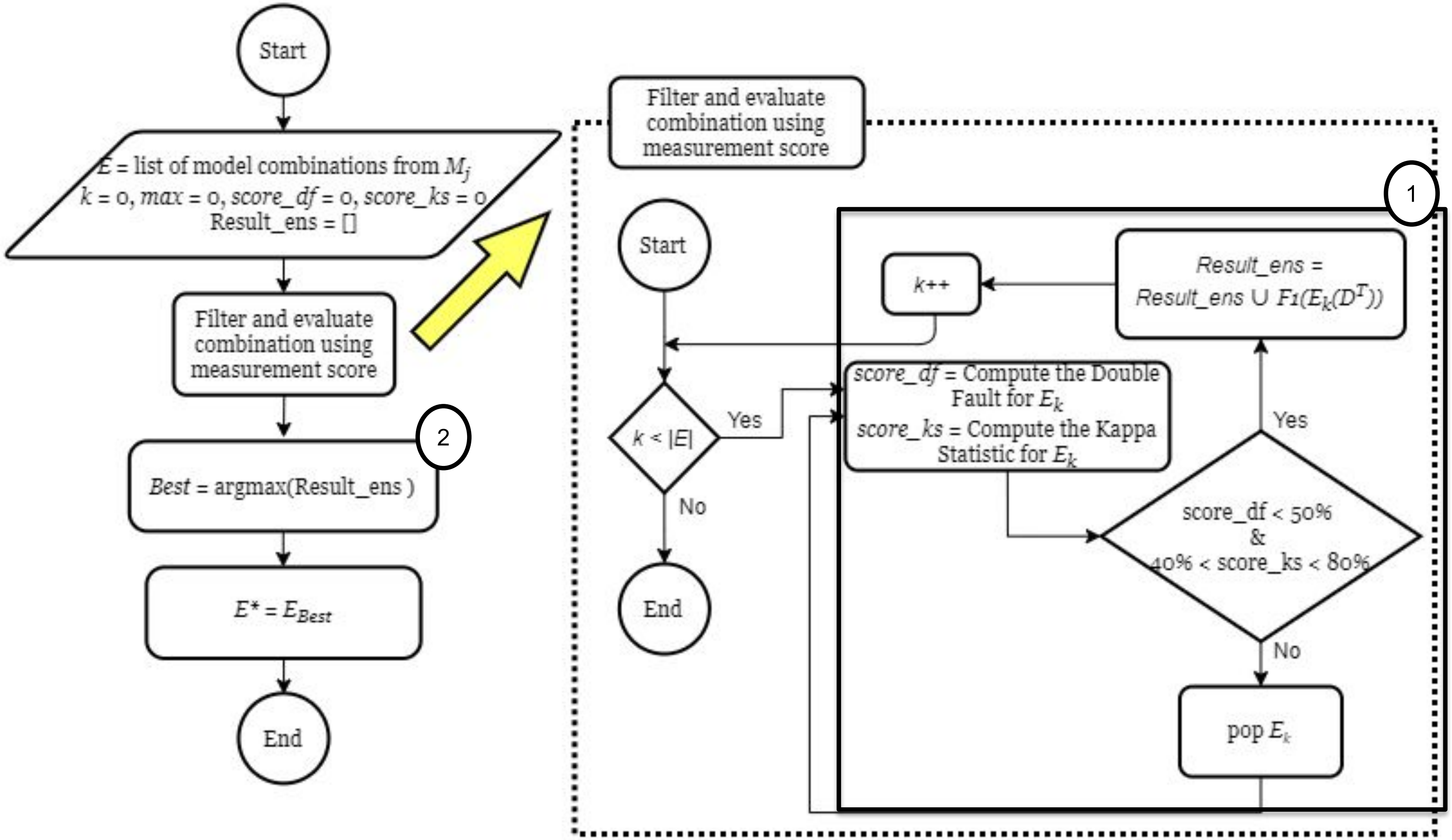
2 Best = argmax([0.99, 0.97])

$M^* = M_0$

Problem: Basic Ensemble Approach Model Creation
Solutions – Double Fault and Kappa Statistics Filter for Ensemble Team

There is 1 loop in this solution:

- Loop by the number of ensemble teams



Example runs

E = ({(M_0, M_1), (M_2, M_3), (M_4, M_5)})
Result_adv = [ ]

Double Fault (E_1) = 90%
Pop E^+_1 from E^+ 1

Double Fault (E_2) = 20%
Kappa Statistics (E_2) = 55%
Append F1 (E_2) to Result_adv 1
Result_adv = [0.62]

Double Fault (E_3) = 10%
Kappa Statistics (E_3) = 60%
Append F1 (E_3) to Result_adv 1
Result_adv = [0.62, 0.60]

Best = argmax([0.62, 0.60]) 2

$E^* = E_{best}$

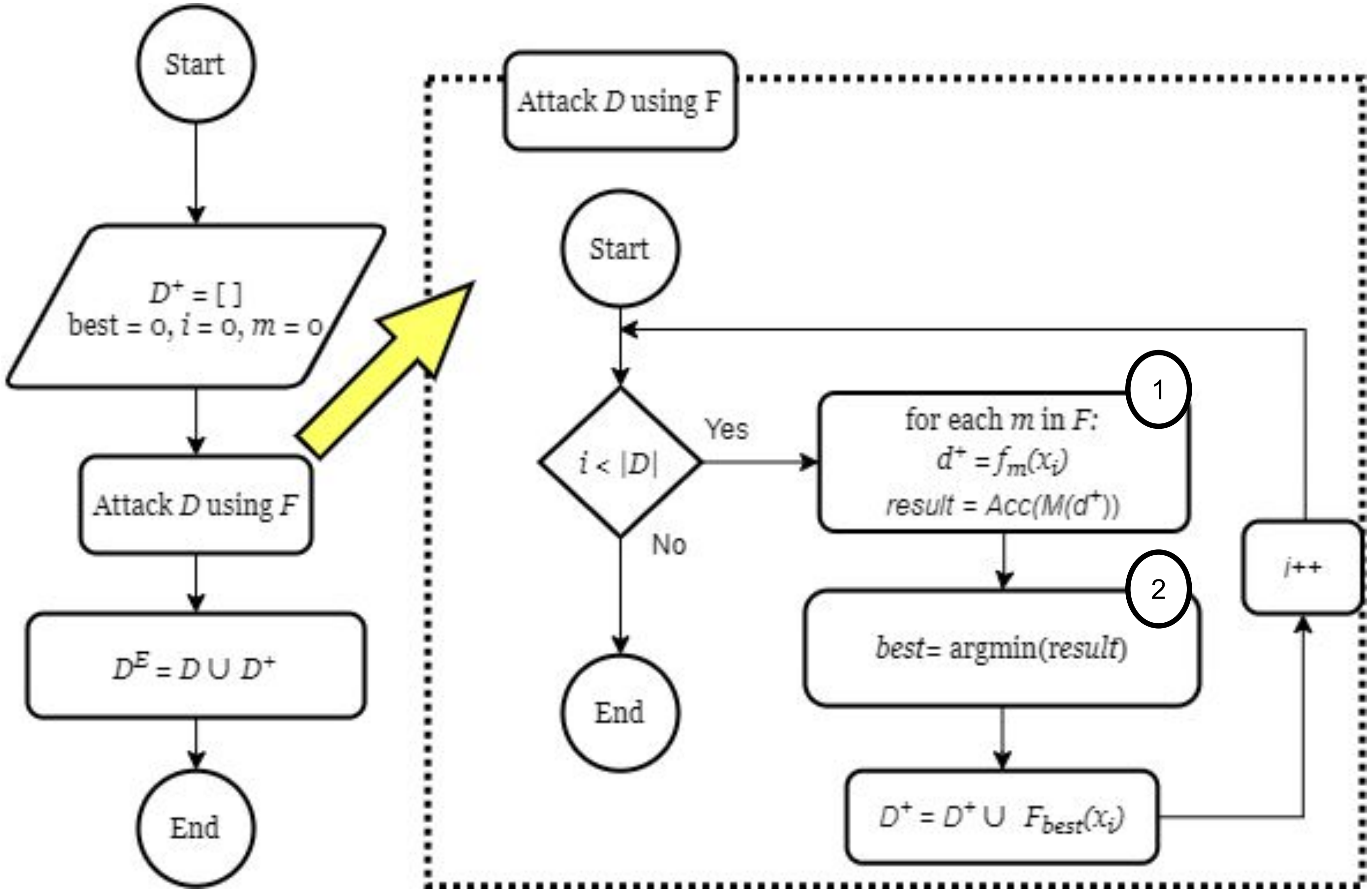
Problem: Adversarial Dataset Generation

Solutions – Exhaustive Comparison

There are 2 loops in this solution:

1st loop by every data x in a aataset D

2nd loop by every adversarial attack technique in F



Example runs

$D^+ = [\ ]$
result = [ ]
F = [FGSM, CW]

↓

1
$d^+ = \text{FGSM}(x_0, y_0)$
Calculate accuracy of d^+
Append the result to result
result = [0.40]

↓

1
$d^+ = \text{CW}(x_0, y_0)$
Calculate accuracy of d^+
Append the result to result
result = [0.40, 0.38]

↓

2 Best = argmin([0.40, 0.38])

↓

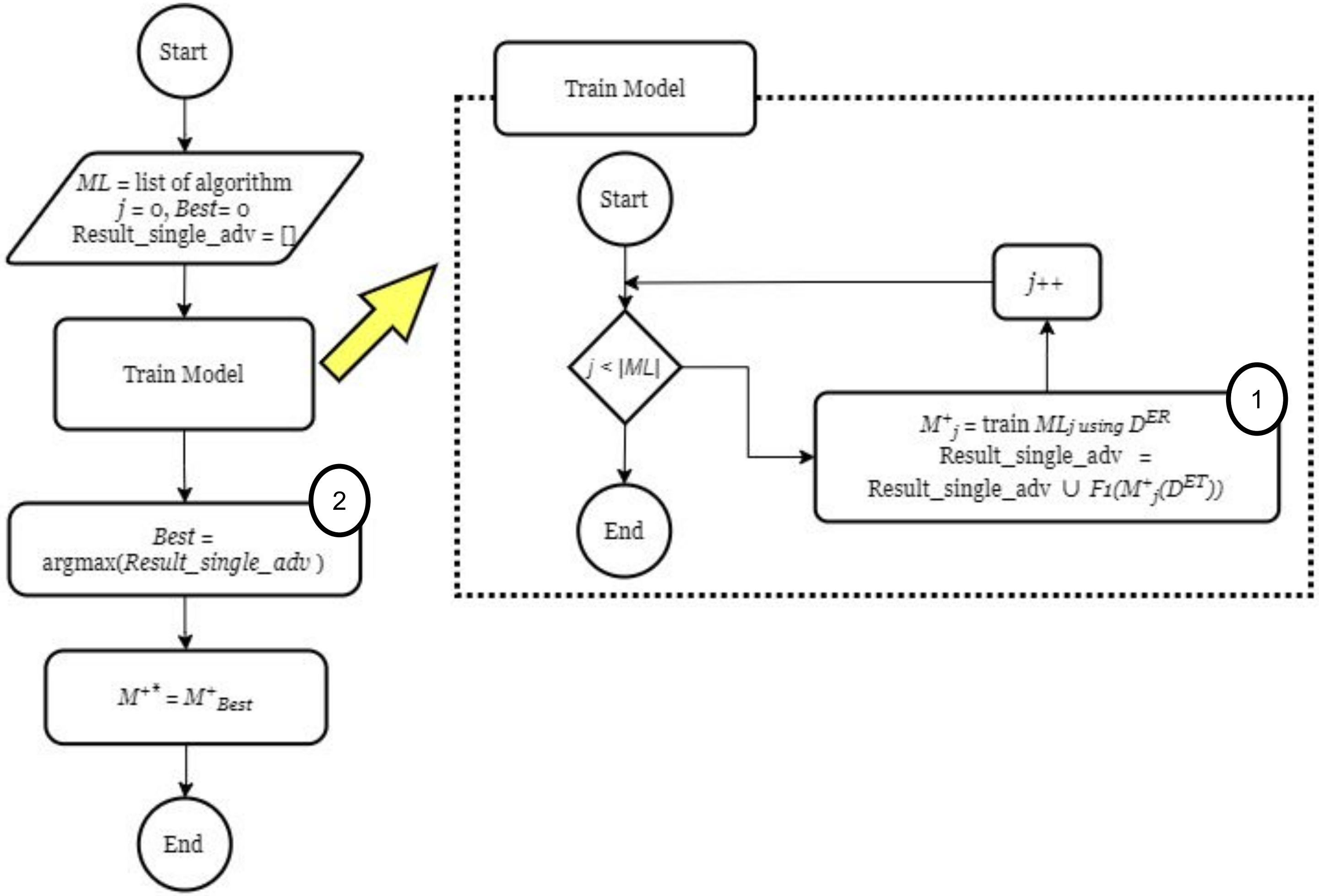
Append the attacked dataset to D^+
$D^+ = [\ F_{best}(x_i)\ ]$

Problem: Adversarial and Ensembled Adversarial Approach

Solutions – F1 Score for Adversarial Model Threshold

There is 1 loop in this solution:

- Loop by the number of machine learning algorithms



Example runs

ML = {DT, LR}
Result_single_adv = [ ]

1 M^{+}_{0} = DT({(x_0, y_0), (x_1, y_1), (x_3, y_3), …, (x_n, y_n)})
Test M^{+}_{0}
Append the result to Result_single_adv
Result_single_adv = [0.99]

1 M^{+}_{1} = LR({(x_0, y_0), (x_1, y_1), (x_3, y_3), …, (x_n, y_n)})
Test M^{+}_{1}
Append the result to Result_single_adv
Result_single_adv = [0.99, 0.97]

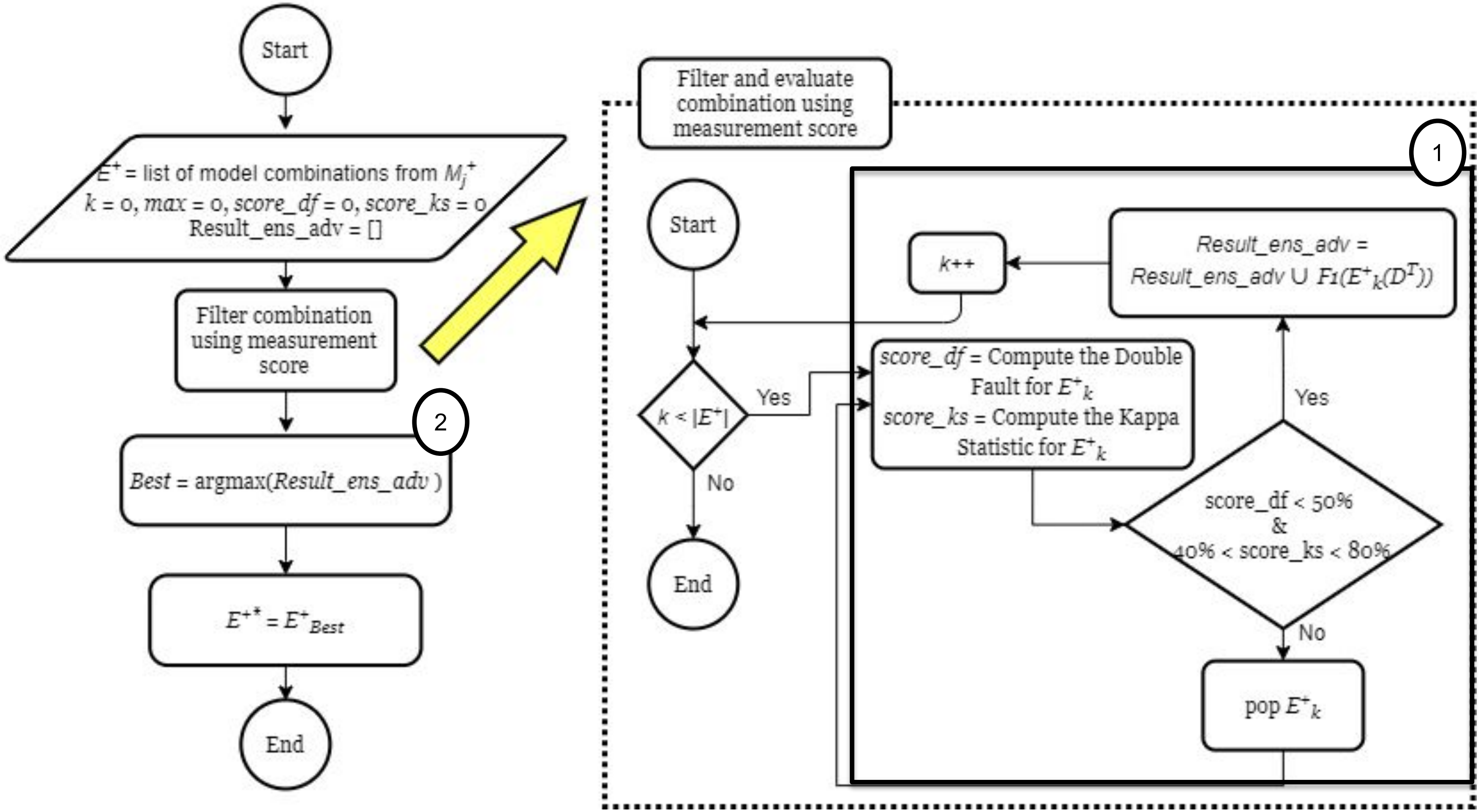
2 Best = argmax([0.99, 0.97])

$M^{+*} = M^{+}_{0}$

Problem: Adversarial and Ensembled Adversarial Approach
Solutions – Double Fault and Kappa Statistics Filter for Ensemble Adversarial Team

There is 1 loop in this solution:

- Loop by the number of ensemble teams



Example runs

$E^+ = (\{(M^+_0, M^+_1), (M^+_2, M^+_3), (M^+_4, M^+_5)\})$
Result_ens_adv = [ ]

Double Fault (E^+_1) = 70%
Pop E^+_1 from E^+ 1

Double Fault (E^+_2) = 30%
Kappa Statistics (E^+_2) = 50% 1
Append F1 (E^+_2) to Result_ens_adv
Result_ens_adv = [0.70]

Double Fault (E^+_3) = 10%
Kappa Statistics (E^+_3) = 60% 1
Append F1 (E^+_3) to Result_ens_adv
Result_ens_adv = [0.70, 0.83]

Best = argmax([0.70, 0.83]) 2

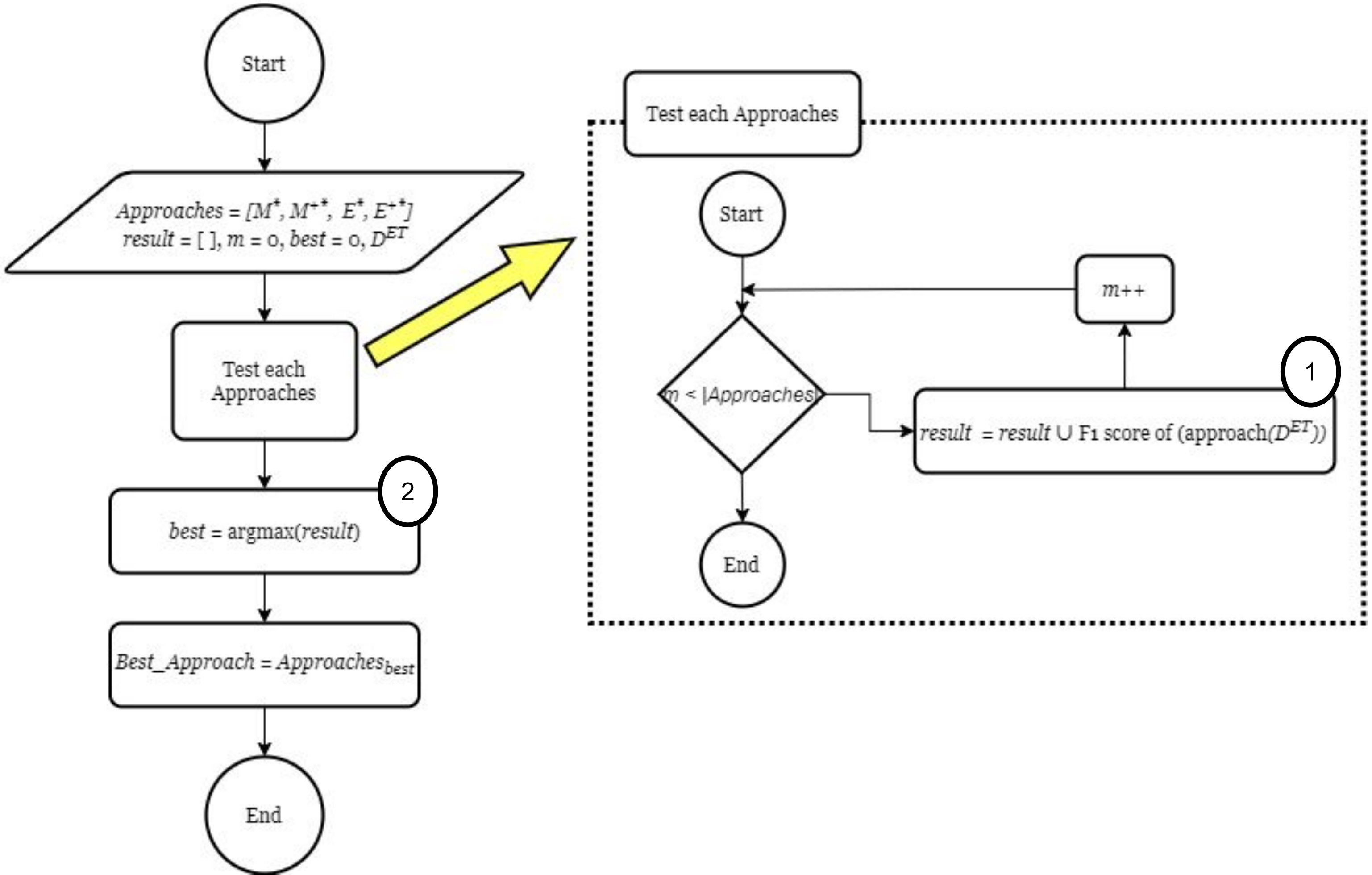
$E^{+*} = E^+_{best}$

Problem: Best Approach

Solutions – Best Approach Selection

There is 1 loop in this solution:

- Loop by the number of approaches available.



Example runs

Approaches = {M^*, M^{+*}, E^*, E^{+*}}
result= [ ]

1 Test M^*
result= [0.40]

1 Test M^{+*}
result= [0.40, 0.50]

1 Test E^*
result= [0.40, 0.50, 0.65]

1 Test E^{+*}
result= [0.40, 0.50, 0.65, 0.82]

2 best = argmax([0.40, 0.50, 0.65, 0.82])

Best_approach = $Approaches_{best}$

Evaluation – Testbed Configuration

Hardware:

Processor : AMD Ryzen 5 3500X 6-Core Processor

RAM : 32 GB

GPU : NVIDIA GeForce RTX 3070

OS : Windows 10

Software:

Library	Version
Jupyter Notebook	6.2.0
Python	3.8.8
Sckit-learn	0.23.2
Numpy	1.18.5
Xgboost	1.3.3
Adversarial Robustness Toolbox	1.6.0

Dataset

- CICIDS 2017

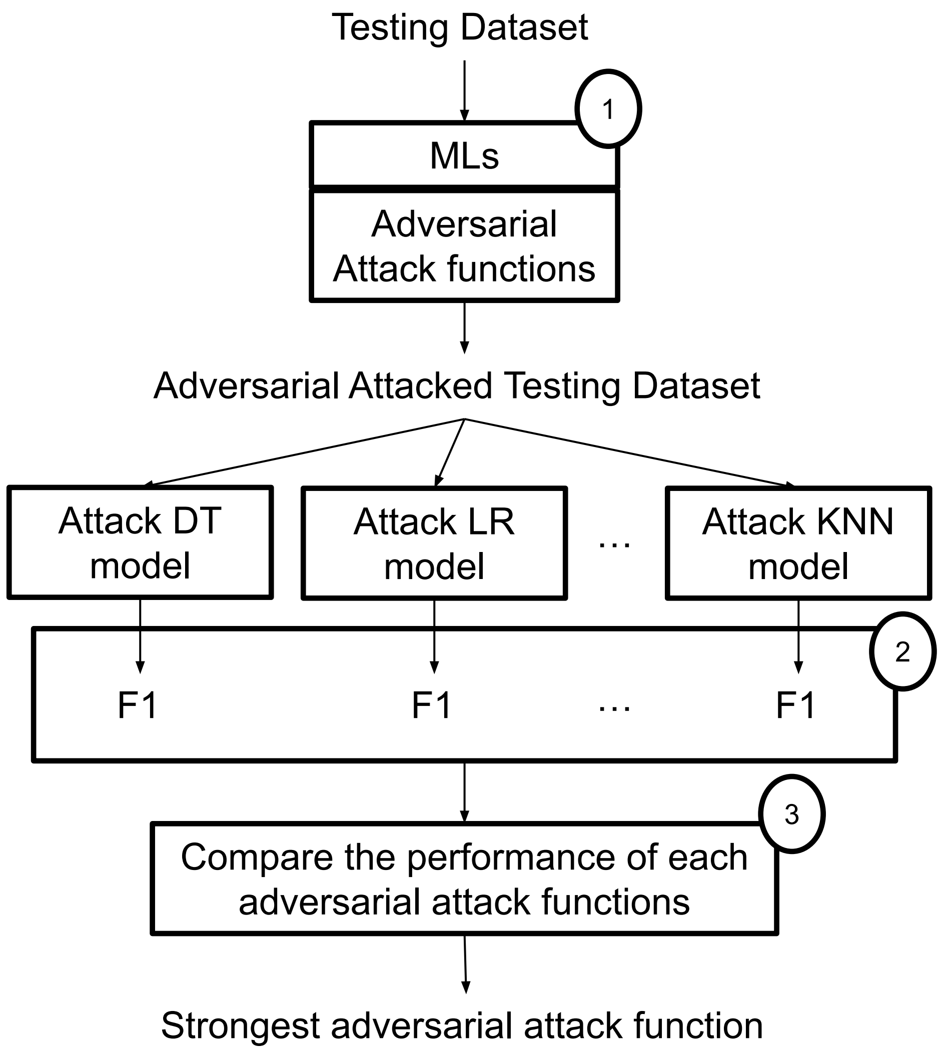
Classifiers

- Decision Tree
- Support Vector Machine
- KNN
- XG Boost
- LR

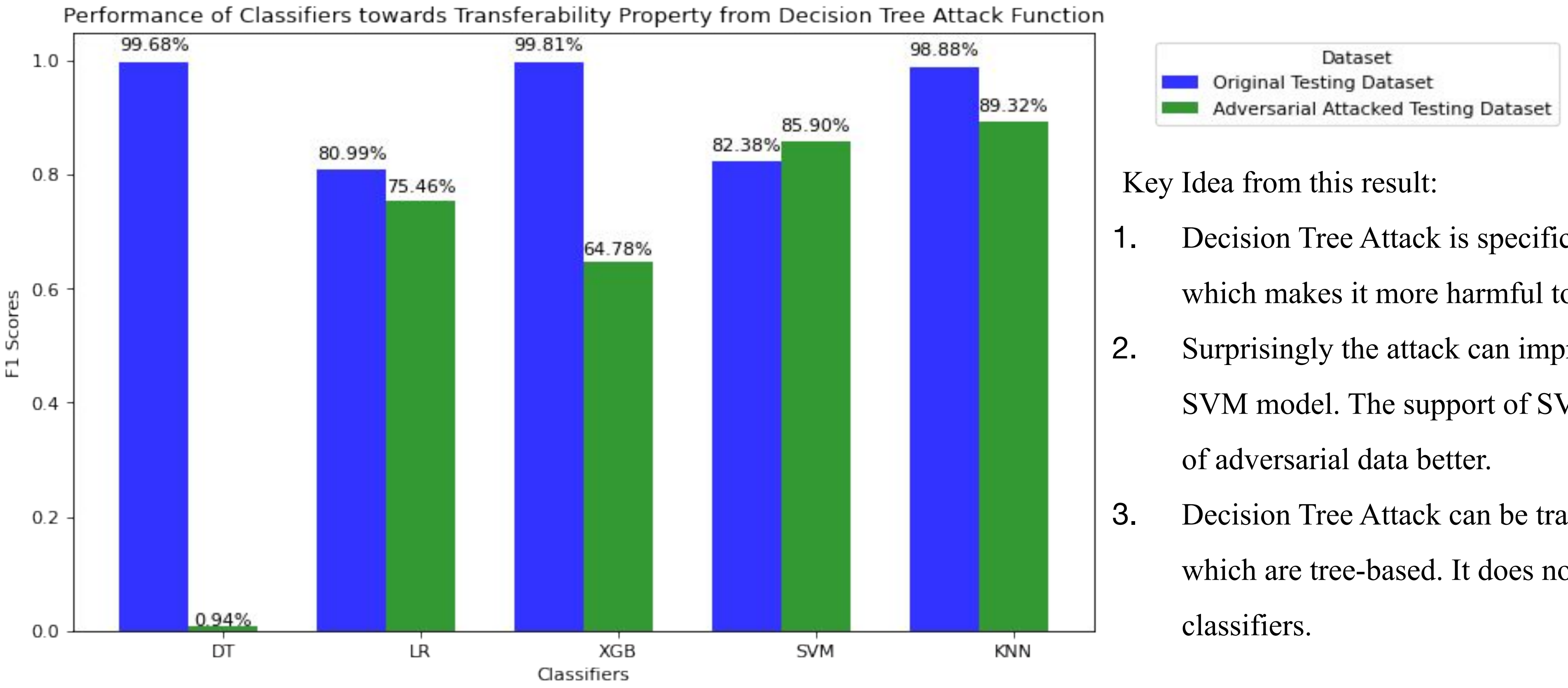
Evaluation – Transferability Property

Steps:

1. Test the transferability property of all possible adversarial attack functions.
2. Compile the performance of all possible tests
3. Conclude the strongest attack function based on the compilation of result from step 2



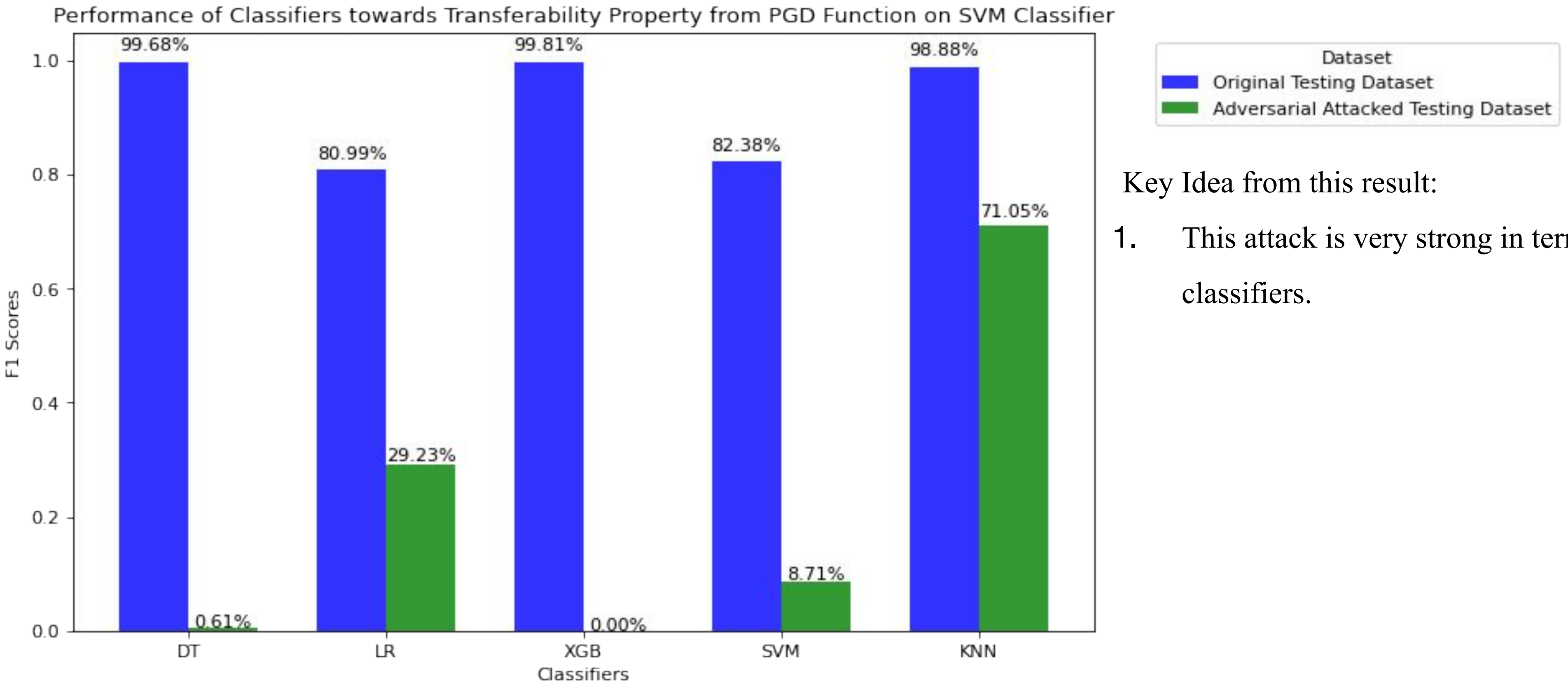
Result – *Decision Tree Attack generated using Decision Tree Classifier*



Key Idea from this result:

1. Decision Tree Attack is specifically made for Decision Tree which makes it more harmful to DT.
2. Surprisingly the attack can improve the performance of SVM model. The support of SVM can divide the distribution of adversarial data better.
3. Decision Tree Attack can be transfer well to classifiers which are tree-based. It does not transfer very well to other classifiers.

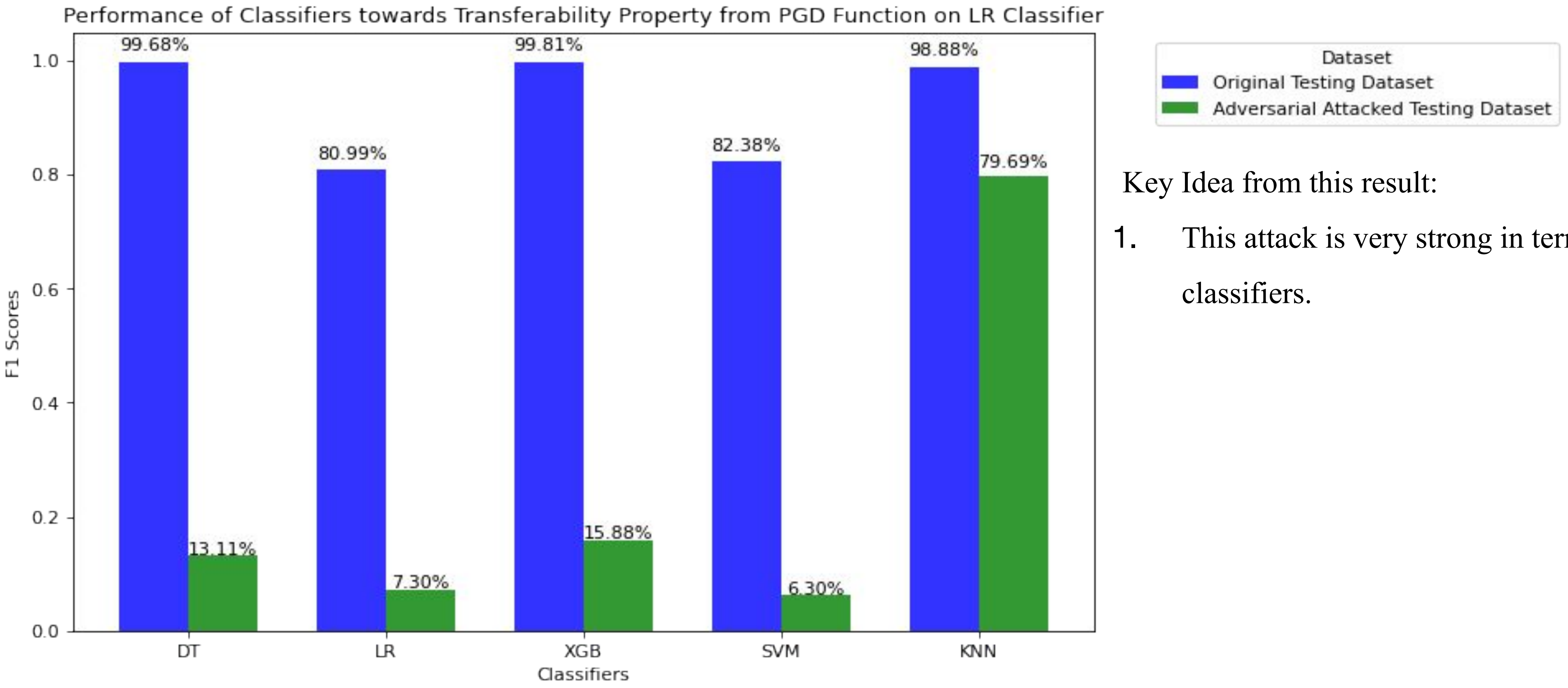
Result – *PGD Attack generated using Support Vector Machine Classifier*



Key Idea from this result:

1. This attack is very strong in terms of attacking other classifiers.

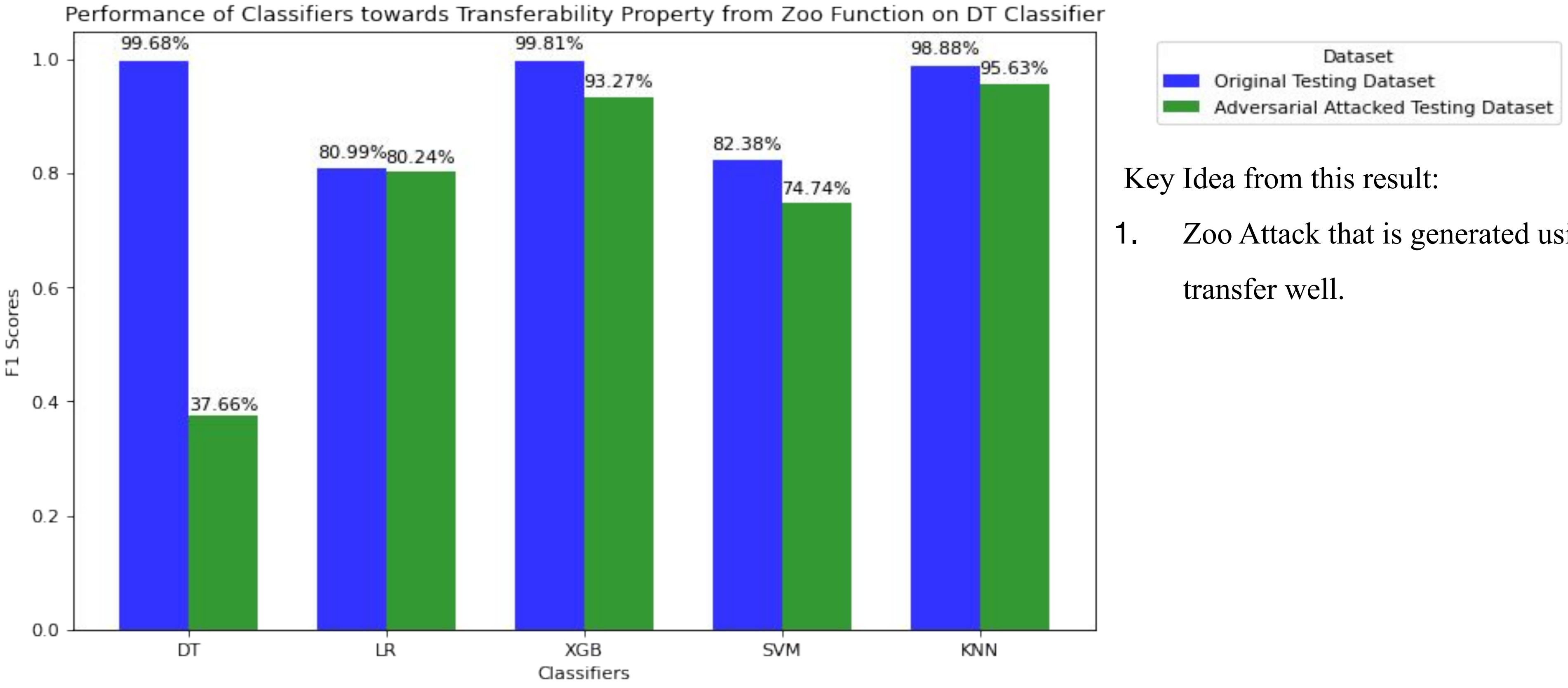
Result – ***PGD Attack*** *generated using* ***Linear Regression Classifier***



Key Idea from this result:

1. This attack is very strong in terms of attacking other classifiers.

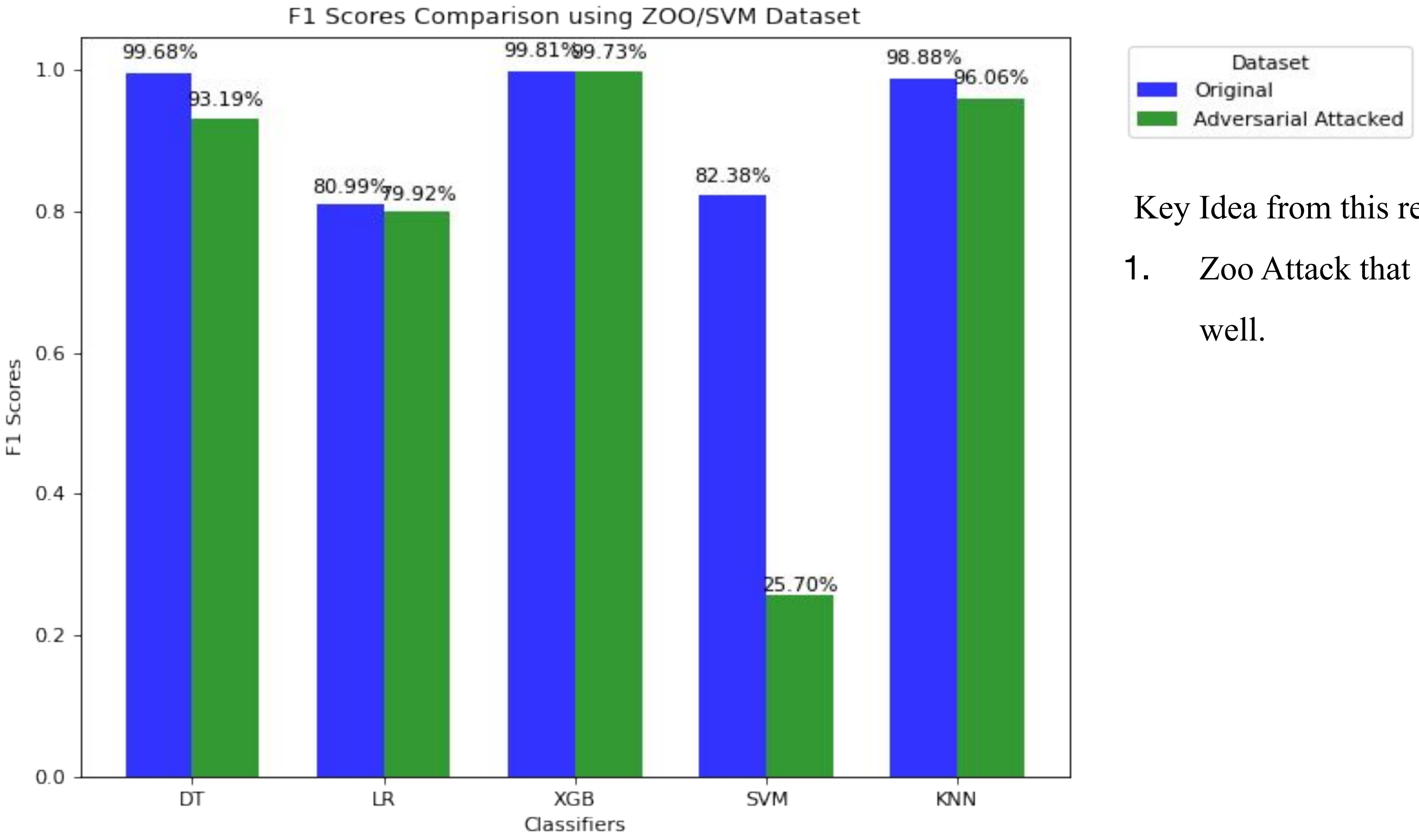
Result – *Zoo Attack generated using Decision Tree Classifier*



Key Idea from this result:

1. Zoo Attack that is generated using decision tree does not transfer well.

Result – ***Zoo Attack*** *generated using* ***SVM Classifier***



Key Idea from this result:

1. Zoo Attack that is generated using SVM does not transfer well.

Evaluation – Adversarial Defense

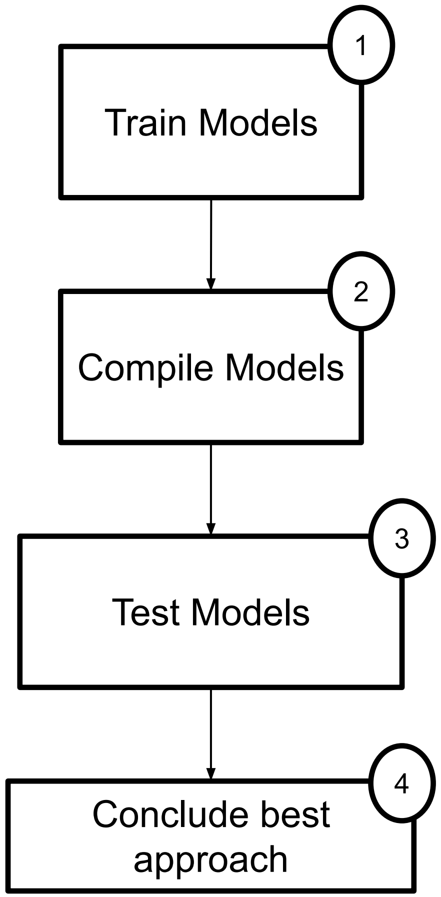
Steps:

1. Train models on:
- clean training dataset
- adversarial attacked dataset

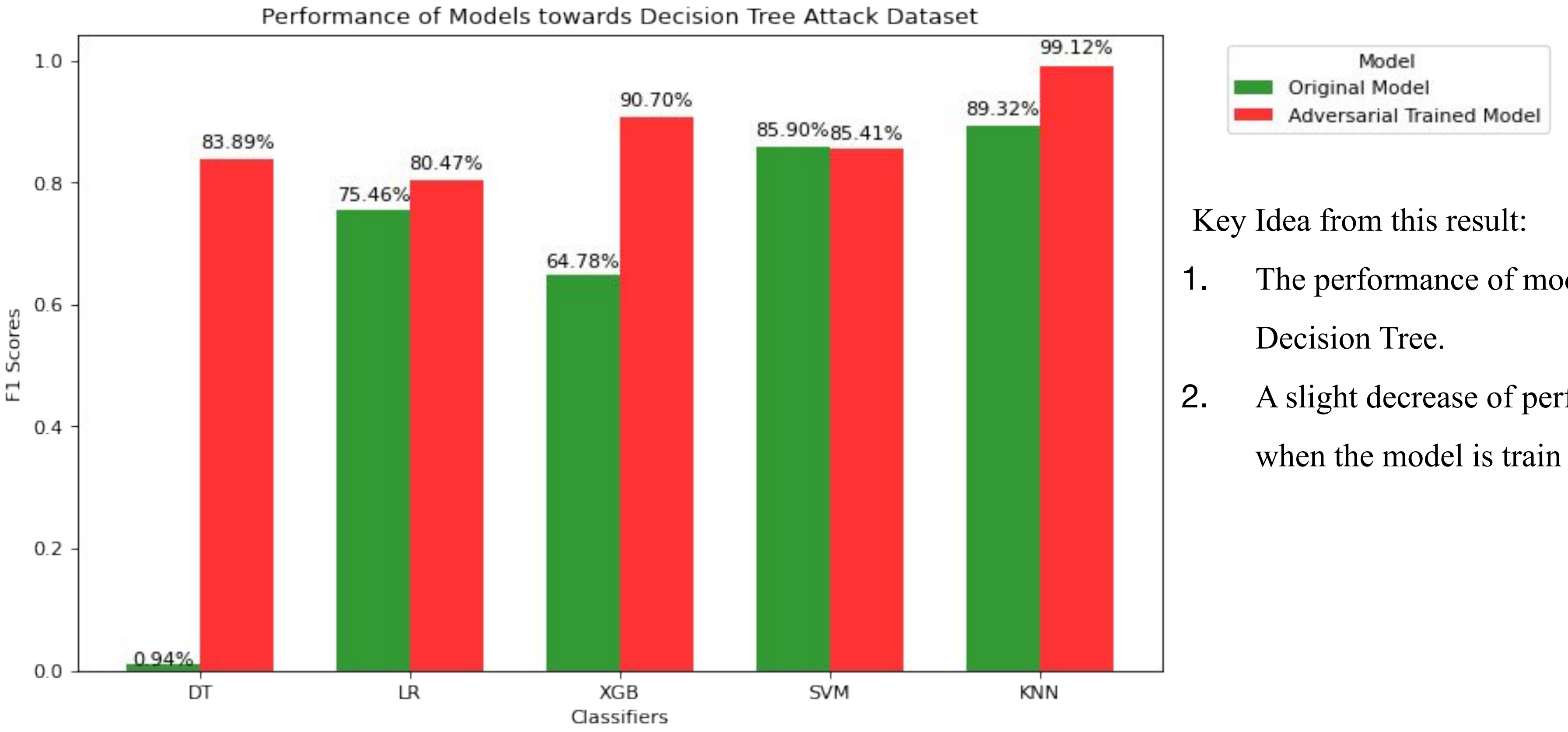
2. Compile the models from:
- step 1 bullet 1 to create an ensemble team.
- step 1 bullet 2 to create an adversarial ensemble team.

3. Test those models on:
- clean test dataset,
- adversarial attacked test dataset and
- It's transferability property.

4. Conclude which approach is the best.



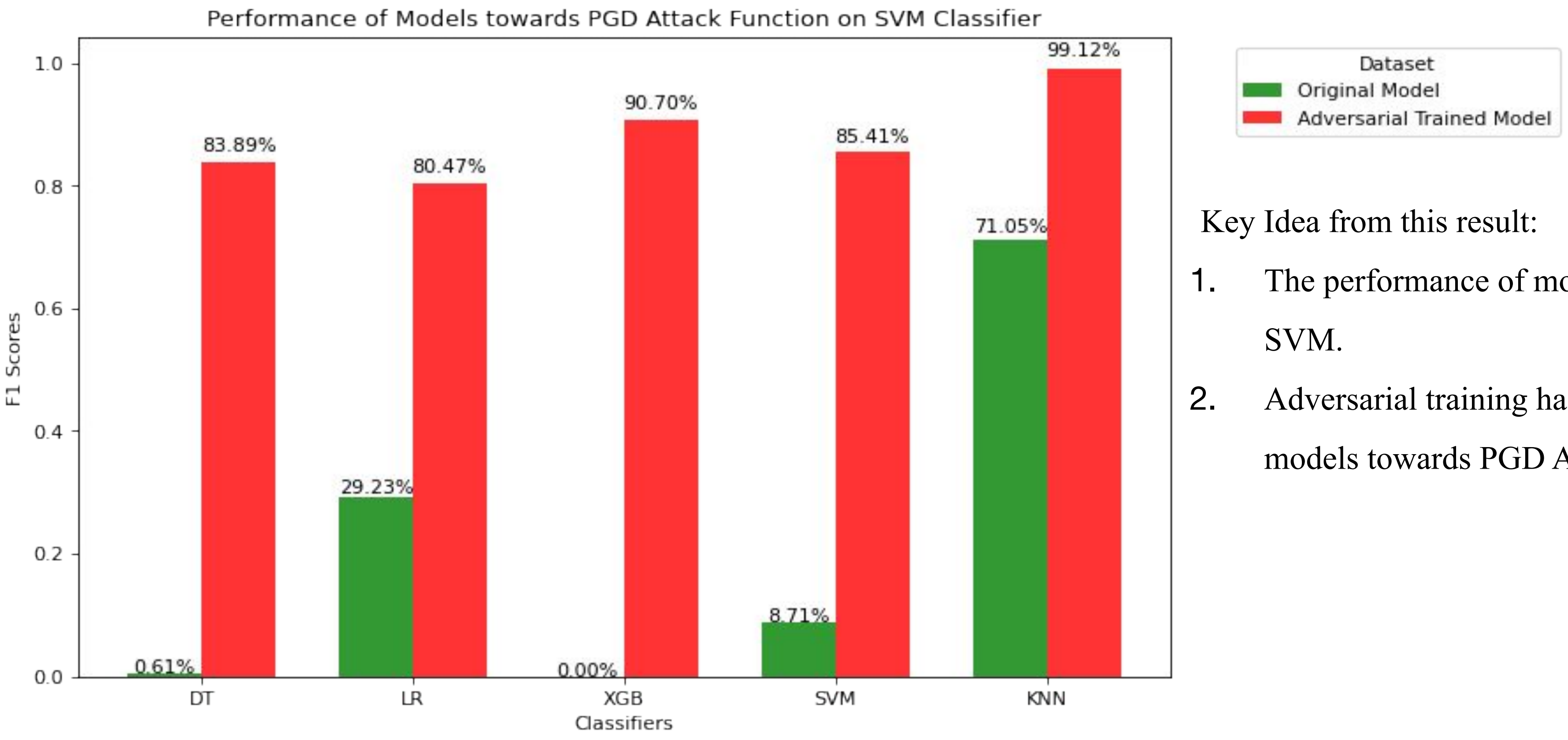
Result – Basic vs. Adversarial on Decision Tree Attack



Key Idea from this result:

1. The performance of model has increase more than 80% for Decision Tree.
2. A slight decrease of performance on the SVM classifier when the model is train using adversarial data.

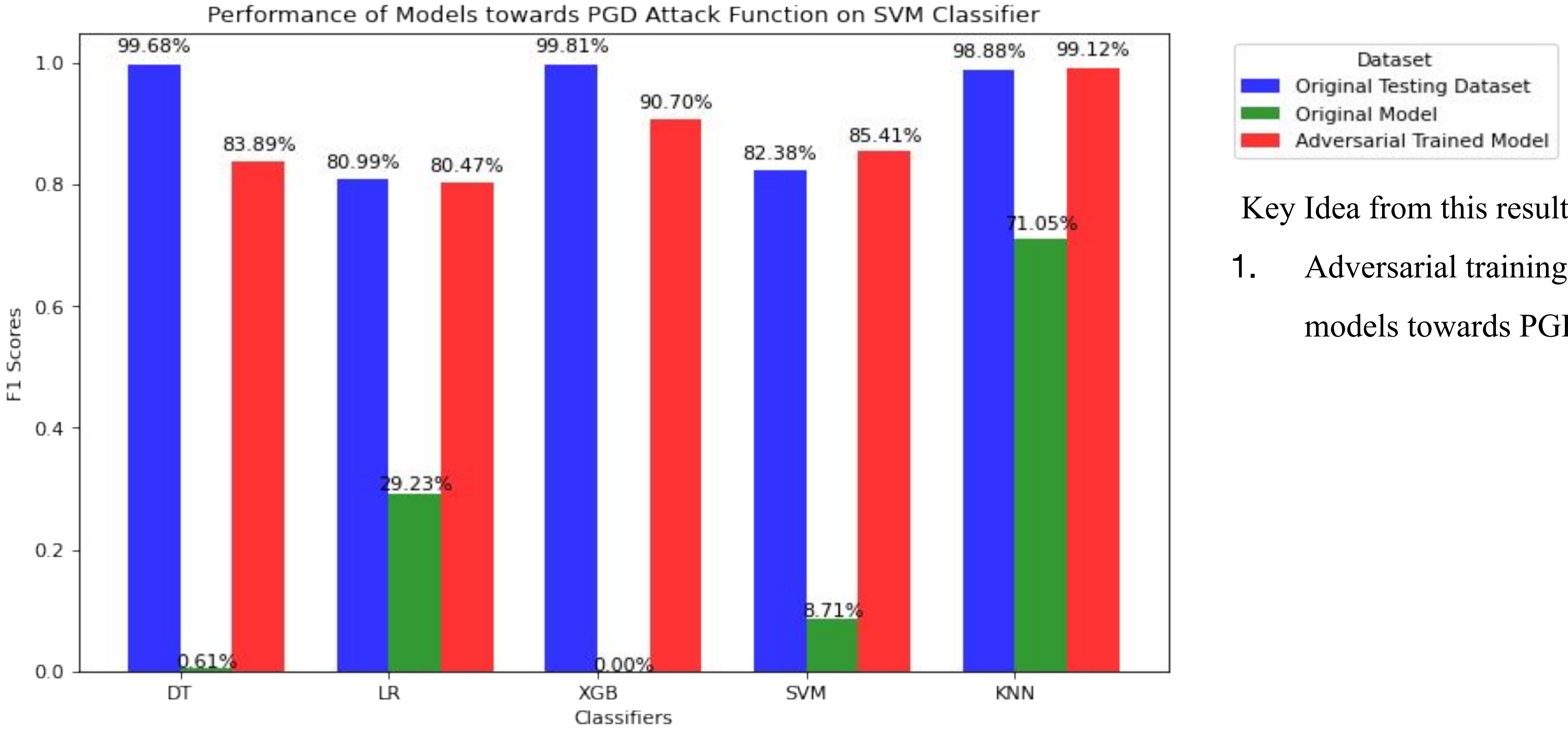
Result – Basic vs. Adversarial on PGD Attack using SVM Classifier



Key Idea from this result:

1. The performance of model has increase more than 70% for SVM.
2. Adversarial training has improve the performance of all models towards PGD Attack using SVM Classifier

Result – Basic vs. Adversarial on PGD Attack using SVM Classifier



Key Idea from this result:

1. Adversarial training has improve the performance of all models towards PGD Attack using SVM Classifier

References (1/7)

[1] Wang, Z. (2018). Deep Learning-Based Intrusion Detection with Adversaries. IEEE Access, 6, 38367–38384. https://doi.org/10.1109/ACCESS.2018.2854599

[2] Pandey, S. (2011). Modern Network Security: Issues and Challenges. International Journal of Engineering Science and Technology, 3.

[3] Martins, N., Cruz, J. M., Cruz, T., & Henriques Abreu, P. (2020). Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. IEEE Access, 8, 35403–35419. https://doi.org/10.1109/ACCESS.2020.2974752

[4] Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., & Roli, F. (2019). Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. Proceedings of the 28th USENIX Security Symposium, 321–338. https://arxiv.org/abs/1809.02861v4

[5] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. https://arxiv.org/abs/1412.6572v3

References (2/7)

[6] Papernot, N., McDaniel, P., & Goodfellow, I. (2016). Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. http://arxiv.org/abs/1605.07277

[7] Miyato, T., Maeda, S. I., Koyama, M., Nakae, K., & Ishii, S. (2016). Distributional smoothing with virtual adversarial training. 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. https://arxiv.org/abs/1507.00677v9

[8] Strauss, T., Hanselmann, M., Junginger, A., & Ulmer, H. (2017). Ensemble methods as a defense to adversarial perturbations against deep neural networks. In arXiv. arXiv. https://arxiv.org/abs/1709.03423v2

[9] Liu, L., Wei, W., Chow, K., Loper, M., Gursoy, E., Truex, S., & Wu, Y. (2019). Deep Neural Network Ensembles Against Deception: Ensemble Diversity, Accuracy and Robustness. In 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) (pp. 274–282). https://doi.org/10.1109/MASS.2019.00040

[10] Gong, Z., Zhong, P., & Hu, W. (2019). Diversity in Machine Learning. IEEE Access, 7, 64323–64350. https://doi.org/10.1109/ACCESS.2019.2917620

References (3/7)

[11] Viera, A. J., & Garrett, J. M. (2005). Understanding interobserver agreement: The kappa statistic. Family Medicine, 37(5), 360–363.

[12] Tang, E., Suganthan, P., & Yao, X. (2006). An analysis of diversity measures. Machine Learning, 65, 247–271. https://doi.org/10.1007/s10994-006-9449-2

[13] Ibitoye, O., Shafiq, O., & Matrawy, A. (2019). Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. IEEE. https://doi.org/10.1109/GLOBECOM38437.2019.9014337

[14] Pawlicki, M., Choraś, M., & Kozik, R. (2020). Defending network intrusion detection systems against adversarial evasion attacks. Future Generation Computer Systems, 110, 148–154. https://doi.org/https://doi.org/10.1016/j.future.2020.04.013

[15] Khamis, R. A., & Matrawy, A. (2020). Evaluation of Adversarial Training on Different Types of Neural Networks in Deep Learning-based IDSs. ArXiv, 0–5. https://doi.org/10.1109/isncc49221.2020.9297344

References (4/7)

[16] Anthi, E., Williams, L., Rhode, M., Burnap, P., & Wedgbury, A. (2020). Adversarial attacks on machine learning cybersecurity defences in industrial control systems. Journal of Information Security and Applications. https://doi.org/10.1016/j.jisa.2020.102717

[17] Apruzzese, G., Andreolini, M., Colajanni, M., & Marchetti, M. (2019). Hardening random forest cyber detectors against adversarial attacks. IEEE Transactions on Emerging Topics in Computational Intelligence. https://doi.org/10.1109/TETCI.2019.2961157

[18] Asadi, M., Jamali, M. A. J., Parsa, S., & Majidnezhad, V. (2020). Detecting botnet by using particle swarm optimization algorithm based on voting system. Future Generation Computer Systems, 107, 95–111. https://doi.org/https://doi.org/10.1016/j.future.2020.01.055

[19] Ghiasi, A., Shafahi, A., & Goldstein, T. (2020). Breaking Certified Defenses: Semantic Adversarial Examples With Spoofed Robustness Certificates. ArXiv, 1–16. https://arxiv.org/abs/2003.08937

[20] Wong, E., Schmidt, F. R., & Zico Kolter, J. (2019). Wasserstein adversarial examples via projected sinkhorn iterations. 36th International Conference on Machine Learning, ICML 2019, 2019-June, 11812–11825. https://arxiv.org/abs/1902.07906v2

References (5/7)

[21] Brendel, W., Rauber, J., Kümmerer, M., Ustyuzhaninov, I., & Bethge, M. (2019). Accurate, reliable and fast robustness evaluation. In arXiv (Issue 1, pp. 1–11). https://arxiv.org/abs/1907.01003v2

[22] Andriushchenko, M., Croce, F., Flammarion, N., & Hein, M. (2020). Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 12368 LNCS, pp. 484–501). https://doi.org/10.1007/978-3-030-58592-1_29

[23] Kotyan, S., & Vargas, D. V. (2019). Adversarial Robustness Assessment: Why both L0 and L∞ Attacks Are Necessary. ArXiv, 1–11. http://arxiv.org/abs/1906.06026

[24] Kurakin, A., Goodfellow, I. J., & Bengio, S. (2019). Adversarial examples in the physical world. 5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings, c, 1–14. https://arxiv.org/abs/1607.02533v4

[25] Papernot, N., Mcdaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. In Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016 (pp. 372–387). https://doi.org/10.1109/EuroSP.2016.36

References (6/7)

[26] Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 2016-Decem, pp. 2574–2582). https://doi.org/10.1109/CVPR.2016.282

[27] Carlini, N., & Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. In Proceedings - IEEE Symposium on Security and Privacy (pp. 39–57). https://doi.org/10.1109/SP.2017.49

[28] Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C. J. (2017). ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models. AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Co-Located with CCS 2017, 15–26. https://doi.org/10.1145/3128572.3140448

[29] Ayub, M. A., Johnson, W. A., Talbert, D. A., & Siraj, A. (2020). Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning. 2020 54th Annual Conference on Information Sciences and Systems, CISS 2020. https://doi.org/10.1109/CISS48834.2020.1570617116

[30] Rigaki, M., & Elragal, A. (2017). Adversarial deep learning against intrusion detection classifiers. CEUR Workshop Proceedings, 2057, 35–48.

References (7/7)

[31] Clements, J., Yang, Y., Sharma, A. A., Hu, H., & Lao, Y. (2019). Rallying adversarial techniques against deep learning for network security. In arXiv. arXiv. https://arxiv.org/abs/1903.11688v1

[32] Warzynski, A., & Kolaczek, G. (2018). Intrusion detection systems vulnerability on adversarial examples. 2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA 2018. https://doi.org/10.1109/INISTA.2018.8466271

[33] Yang, K., Liu, J., Zhang, C., & Fang, Y. (2019). Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. Proceedings - IEEE Military Communications Conference MILCOM, 2019-October, 559–564. https://doi.org/10.1109/MILCOM.2018.8599759

[34] Peng, Y., Su, J., Shi, X., & Zhao, B. (2019). Evaluating deep learning based network intrusion detection system in adversarial environment. ICEIEC 2019 - Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication, 61–66. https://doi.org/10.1109/ICEIEC.2019.8784514

[35] Holscher, E., Johnson, A., & Kaufmann, M. (2021). IBM-ART. Retrieved from https://adversarial-robustness-toolbox.readthedocs.io/en/stable/modules/attacks/evasion.html#boundary-attack-decision-based-attack