Talk at SKKU in 2023

Adversarial Machine Learning in Network Intrusion Detection

Dr Dongseong “Dan” Kim (김동성)

Associate Professor (≈ tenured full professor in the US),

The University of Queensland (UQ), Brisbane, Australia

E-mail: dan.kim@uq.edu.au

Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

UQ – A **top 50** global university

33

- **NTU Performance Ranking of Scientific Papers**
- **2023**

36

***U.S. News* Best Global Universities**

2023

50

- **QS World University Ranking 2023**

47

Academic Ranking of World Universities

2023

53

***Times Higher Education* World University Ranking 2023**

https://www.uq.edu.au/research/about/international-rankings

An Interdisciplinary Cyber Security Group

Industry

AusCERT
and members

The University Of Queensland
AUSTRALIA
CREATE CHANGE

Information Technology Services Division

Industry, Governments & International Organisations

Research

Faculties:

- Engineering, Architecture and Information Technology
- Science (Maths and Physics)
- Humanities and Social Sciences (HASS)
- Business, Economics and Law (BEL)
- Medicine

Centre for Policy Futures

Institute for Social Science Research

AusCERT Research

Teaching

CPD Courses

Bachelor of Computer Science
(Cyber Security Major)

Master of Cyber Security
Grad Dipl in Cyber Security
(4 Fields of Study)
Grad Cert in Cyber Security

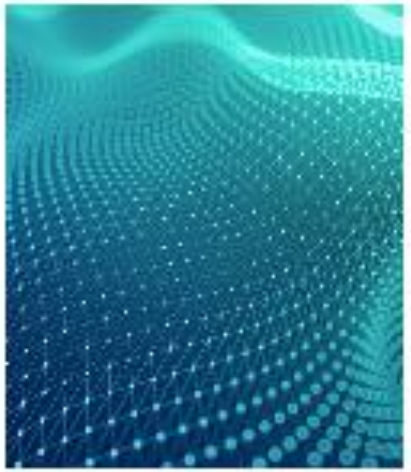
PhD & MPhil

Interdisciplinary Research

Secure quantum communications

Secure communications for space

Cyber autonomy and automation



Data privacy and user data control

Cyber law and ethics

Secure software engineering

National security and cyber policies

Cyber criminology

My Education, Research and Funding

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Master's, 2001–2003

- Crypto HW design
- AI for Network Security

PhD, 2003–2008

- Security & Privacy for Sensor Nets

Postdoc, 2008–2011

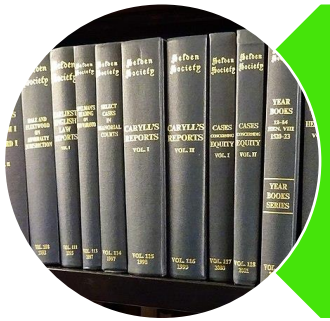
- Security and Dependability

Academic, 2011 – present.

- Cybersecurity

Research outcomes summary

Grants secured: approx. total 2.3M USD



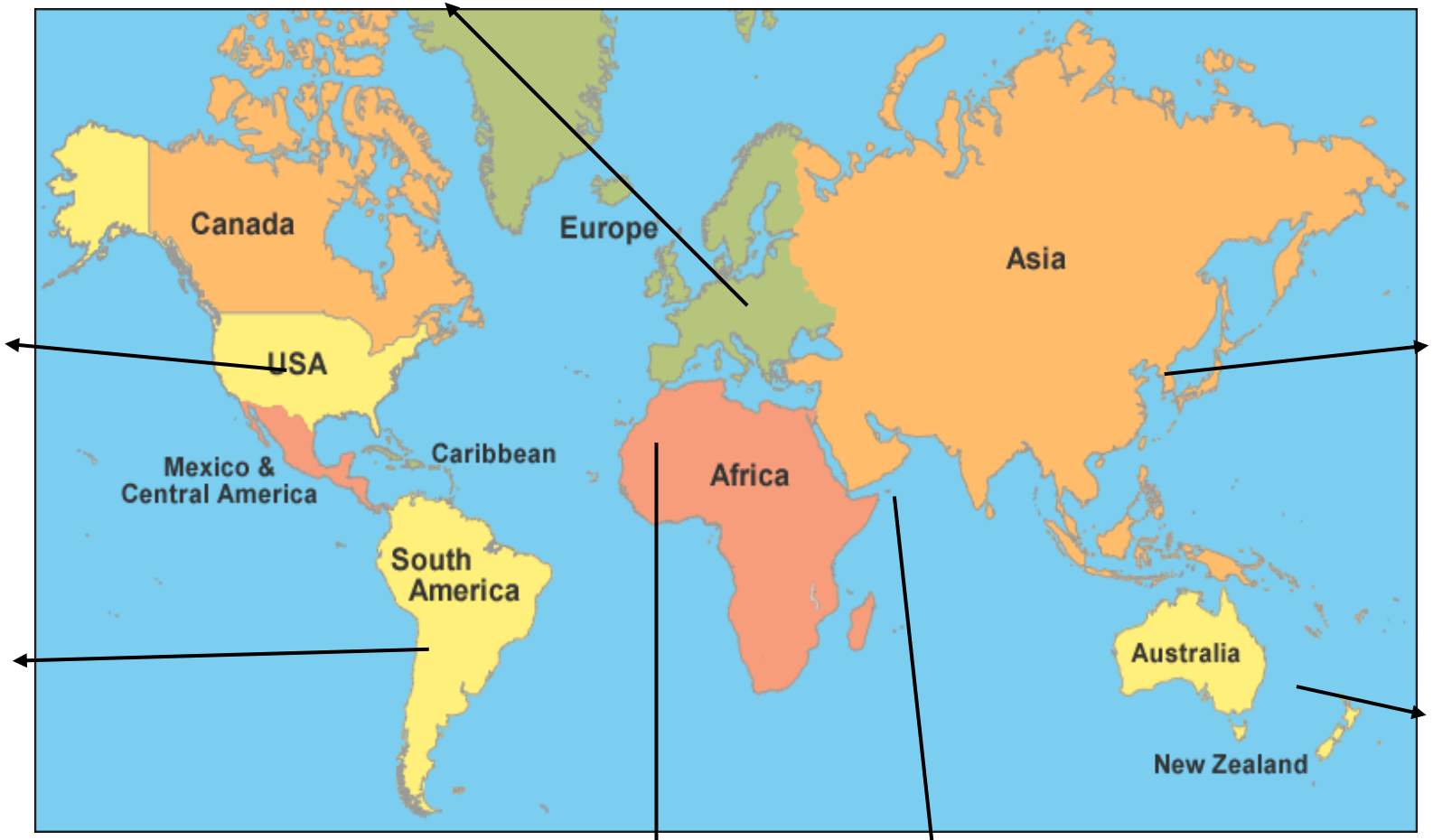
160+ research papers

5600+ Google citations

Funded by South Korea, NATO SPS, USA, Japan, NZ, Qatar, Australia

International Collaborators

- Neeraj Suri (Lancaster, U.K)
- Armin Zimmermann (TU L, Germany)
- R. Natella (USNFII, Italy)



- K. Trivedi (Duke U)
- J. Cho (Virginia Tech)
- D. Huang (Arizona State U.)
- US ARL

- P. Maciel (FUPR, Brazil)
- E. Andrade (FRUP, Brazil)

Korea
- H. Lim (Kentech)
- Y. Paek (SNU)
- HK. Kim (Korea U)
- HS. Kim (SKKU)
- J. Park (KAU)

Japan
- F. Machia (Tsukuba)

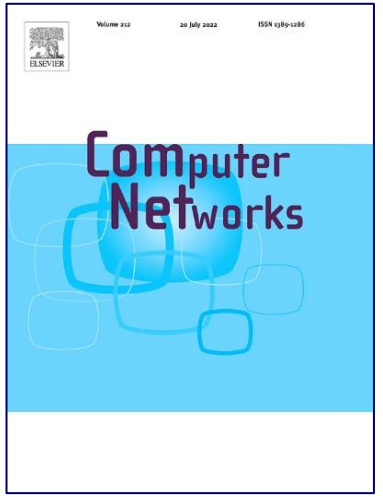
AUS/NZ
- J. Jaccard (Massey)
- I. Welch (VUW)
- J. Hong (UWA)
- …

- A. Haqiq (H1U, Morocco)

- K. Khan, A Nhlabatsi, N Fetais (QU, Qatar)

Professional activities (Dan Kim)

- Editorial Board Member
  - Associate Editor, IEEE Communications Surveys and Tutorials (impact factor: 25.25 (2021), #1 impact factor among all the IEEE journals), 2021 - present.
  - Editorial Board Member, Elsevier Computers and Security (impact factor: 4.438), 2019 - present.
  - Editorial Board Member, Elsevier Computer Networks (impact factor: 4.474), April 2022 - present.



- General (co-)chair for conferences
  - The **54th** IEEE/IFIP Int. Conf. on Dependable Systems and Networks (**DSN 2024**) to be held in Brisbane, Australia.
  - The 24th Australasian Conf. on Information Security and Privacy (ACISP 2019)
  - The 22nd IEEE Pacific Rim Int. Sym. on Dependable Computing (PRDC 2017)

My Research areas: Cyber **G.A.M.E**

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Graphical Security Models (GSM):	• Model-based Cyber Security Risk Analysis
AI for Cybersecurity & Cyber Security for AI:	• Securing AI systems and Cybersecurity using AI techniques
Moving Target Defence (MTD):	• Resilient and Proactive Cyber Defence
Evolving Attacks and Defense Automation:	• Red team and Blue team Automation & Evaluation

My Research areas: Cyber **G.A.M.E – PhD students**

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Graphical Security Models (GSM):

- IoT: Kok Onn Chee (PhD students, UQ)
- Vehicle nets: Nhung Nguyen (PhD students, UQ)

AI for Cybersecurity & Cyber Security for AI:

- AML for IoT: Ke He (PhD student, U Auckland)
- AML for Networks: Subrat Swain (PhD student, UQ/IITD)
- AML for vehicles: Isha Pali (PhD student, UQ/IITD)
- ML/DL for Distributed IDS: Ulysses Lam (PhD student, UQ)

Moving Target Defence (MTD):

- Against AI powered attacks: Tina Moghaddam (PhD student, UQ)

Evolving Attacks and Defense Automation:

- Red team automation using DRL: William Li (PhD student, UQ)

AI for cyber security & cyber security for AI

- Q1: What/how can we use AI for Cybersecurity?
- Q2: How can we make AI secure/robust?

My recent R&D project related to AI

Project Title	Funded by	Amount	Years	Role
Resilient Learning-based Defense in Adversarial Uncertain Environments	The USA RDECOM International Technology Center - Pacific (ITC-PAC) and the US Army Research Lab (ARL)	420K USD	2020-2023	PI
Cyber Defenses and Their Assessment for Resilient Autonomous Systems (to be awarded)	The USA RDECOM International Technology Center - Pacific (ITC-PAC) and the US Army Research Lab (ARL)	420K USD	2023-2026	Co-PI

- ML/DL based IDS for SDN/vehicle nets
- Continual learning
- Federated learning
- Intrusion response system (IRS)

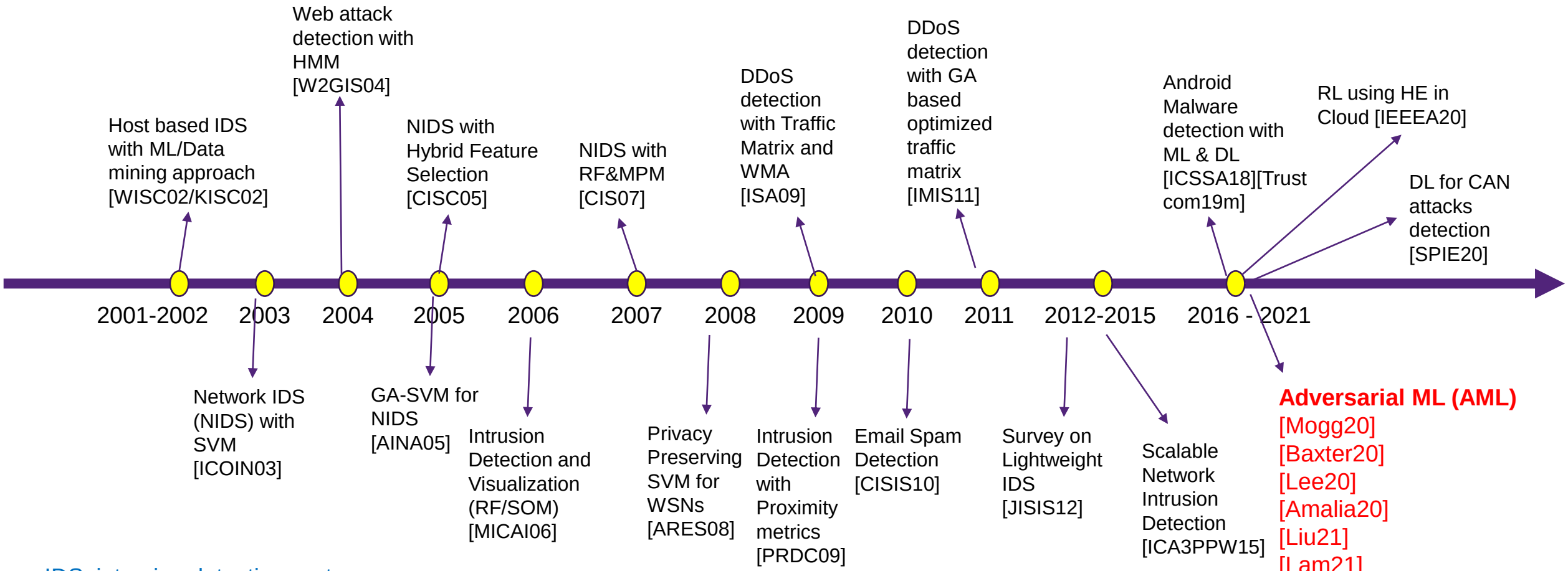
My experience in AI (Machine/Deep learning)

- Worked on ML for cybersecurity since **2001**
  - Master thesis (Machine Learning techniques for Network Intrusion Detection) in 2003
  - A part of PhD thesis - Privacy preserving data mining techniques for Sensor Networks in 2008

- Published research papers in ML/DL for cybersecurity
  - Host/Network Intrusion Detection
  - DDoS attacks detection
  - Spam (e-mail) detection
  - Android malware detection
  - AI for cyberattacks and defense automation

- Security for AI
  - AML in images
  - AML for IDS
  - Privacy for ML/DL

My research on AI security & Security for AI (2001-present)



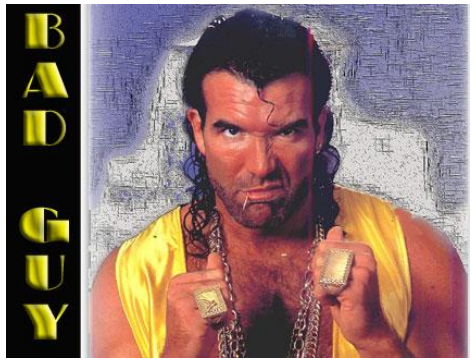
- IDS: intrusion detection system
- ML: Machine Learning
- SVM: support vector machines
- GA: Genetic Algorithm
- RF: Random Forest
- SOM: Self Organizing Map
- MPM: Minimax Probability Machine
- WMA: Weighted Moving Average

Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

Definition of Intrusion and IDS

- Intrusion
  - A set of actions that attempt to compromise the, confidentiality, integrity or availability of computing resources via
    - Causing Denial of Service
    - Creating Backdoor(Trojan Horse)
    - Planting Viruses
    - Exploiting Software Vulnerability



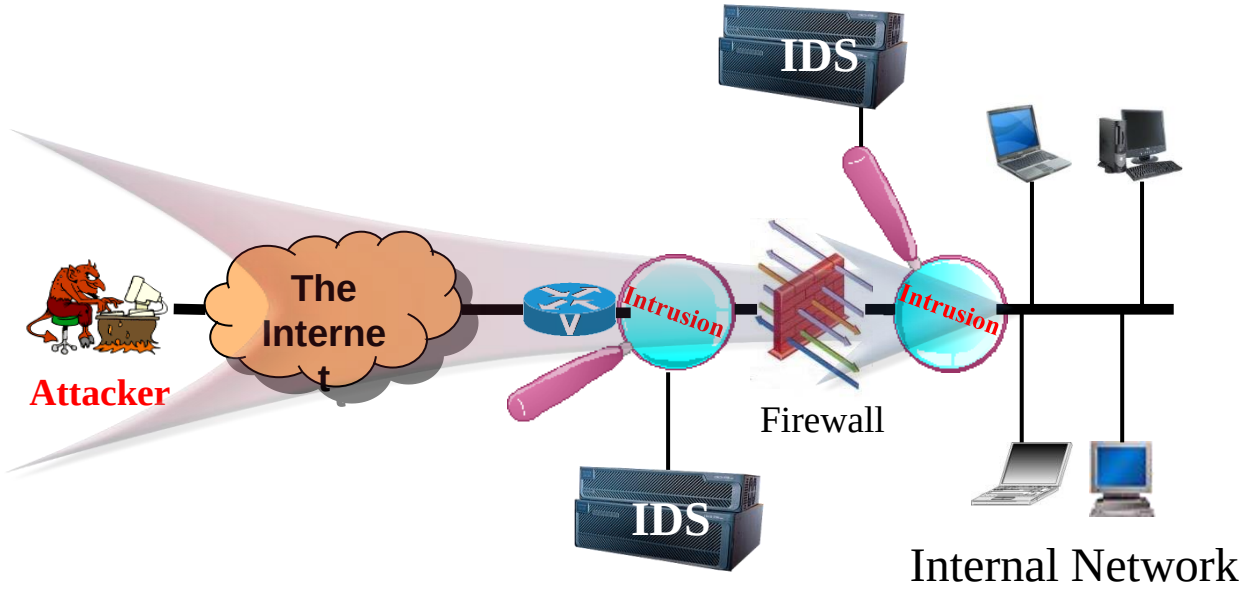
[**Anderson**,1980]

Intrusion Detection

- The process of monitoring and analyzing the events occurring in a computer and/or network system in order to detect signs of security problems
- Primary assumption: Certain (e.g., user, program activities, networks) can be monitored and modeled
- Steps
  1. Monitoring and analyzing host/network
  2. Identifying misuse/abnormal activities
  3. Assessing severity and raising alarm

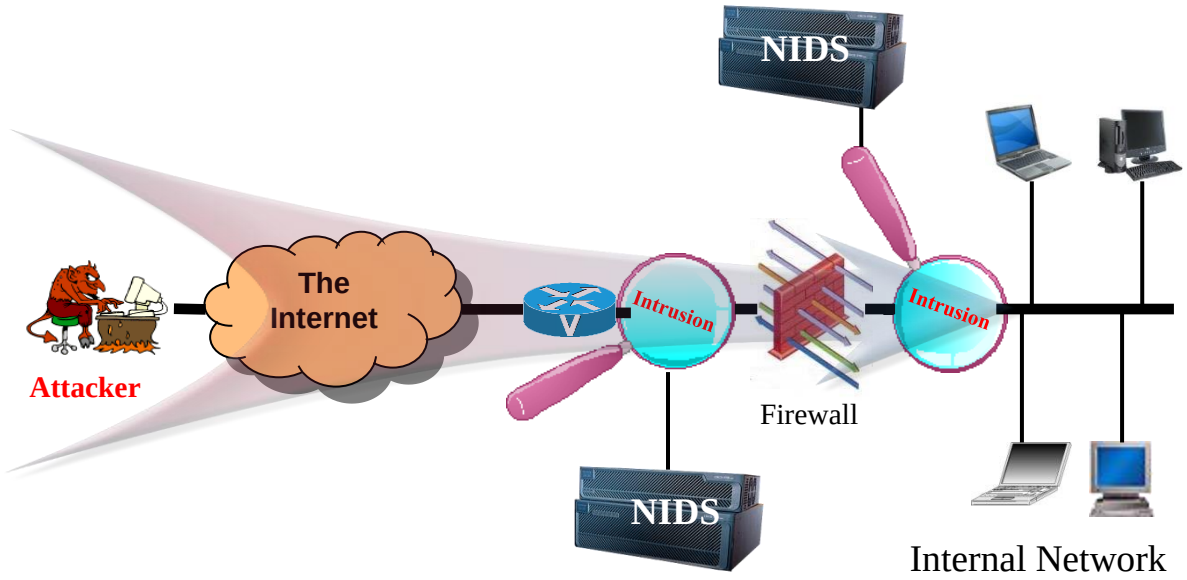
Intrusion Detection System (IDS)

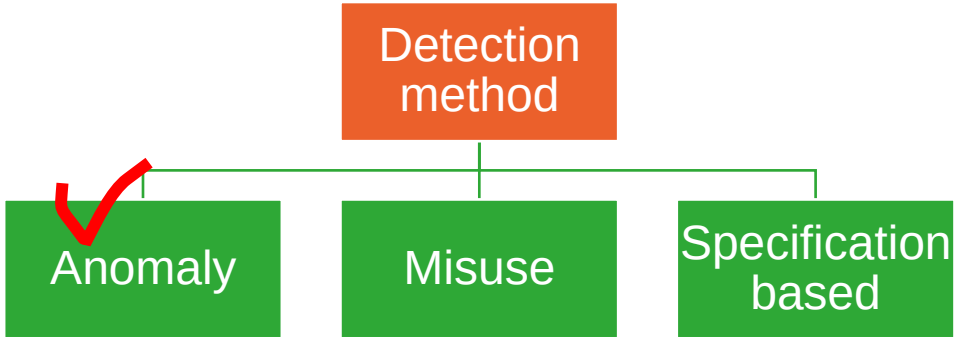
- combination of software and hardware that attempts to perform “intrusion detection”
- raise the alarm, when possible, intrusion or suspicious patterns are observed

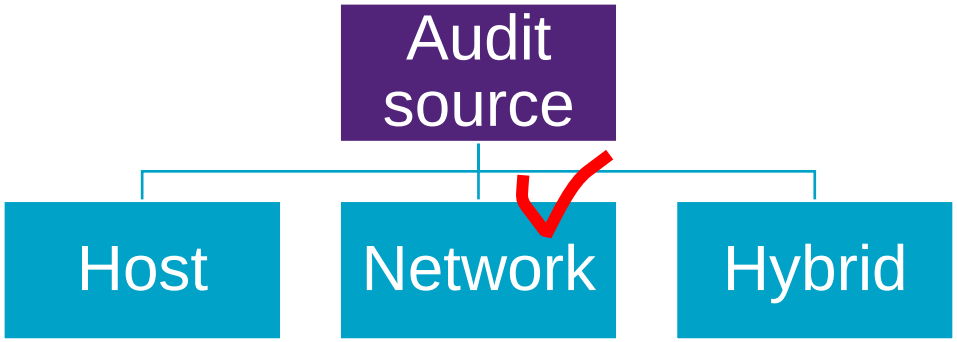


J. Song et al. A Proposal of New Benchmark Data to Evaluate Mining Algorithms for Intrusion Detection, 23rd Asia Pacific Advanced Network Meeting, 2007

What is an Intrusion Detection System (IDS)?







KDD 1999 Dataset features

	feature name	description	type
1	duration	length (number of seconds) of the connection	continuous.
2	protocol_type	type of the protocol, e.g. tcp, udp, etc.	symbolic.
3	service	network service on the destination, e.g., http, telnet, etc.	symbolic.
4	flag	normal or error status of the connection	symbolic.
5	src_bytes	number of data bytes from source to destination	continuous.
6	dst_bytes	number of data bytes from destination to source	continuous.
7	land	1 if connection is from/to the same host/port; 0 otherwise	symbolic.
8	wrong_fragment	number of ``wrong'' fragments	continuous.
9	urgent	number of urgent packets	continuous.
10	hot	number of ``hot'' indicators	continuous.
11	num_failed_logins	number of failed login attempts	continuous.
12	logged_in	1 if successfully logged in; 0 otherwise	symbolic.
13	num_compromised	number of ``compromised'' conditions	continuous.
14	root_shell	1 if root shell is obtained; 0 otherwise	continuous.
15	su_attempted	1 if ``su root'' command attempted; 0 otherwise	continuous.
16	num_root	number of ``root'' accesses	continuous.
17	num_file_creations	number of file creation operations	continuous.
18	num_shells	number of shell prompts	continuous.
19	num_access_files	number of operations on access control files	continuous.
20	num_outbound_cmds	number of outbound commands in an ftp session	continuous.
21	is_host_login	1 if the login belongs to the ``hot'' list; 0 otherwise	symbolic.
22	is_guest_login	1 if the login is a ``guest''login; 0 otherwise	symbolic.

NSL-KDD data feature

F#	Feature name	F#	Feature name	F#	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

https://www.unb.ca/cic/datasets/nsl.html

The UNSW-NB15 Dataset

The University Of Queensland AUSTRALIA

ID	Feature	ID	Feature	ID	Feature
1	attack_cat	16	dloss	31	response_body_len
2	dur	17	sinpkt	32	ct_srv_src
3	proto	18	dinpkt	33	ct_state_ttl
4	service	19	sjit	34	ct_dst_ltm
5	state	20	djit	35	ct_src_dport_ltm
6	spkts	21	swin	36	ct_dst_sport_ltm
7	dpkts	22	stcpb	37	ct_dst_src_ltm
8	sbytes	23	dtrcpb	38	is_ftp_login
9	dbytes	24	dwin	39	ct_ftp_cmd
10	rate	25	tcprtt	40	ct_flw_http_mthd
11	sttl	26	synack	41	ct_src_ltm
12	dttl	27	ackdat	42	ct_srv_dst
13	sload	28	smean	43	is_sm_ips_ports
14	dload	29	dmean		
15	sloss	30	trans_depth		

The UNSW-NB15 Dataset

Types of attacks	Testing dataset		Training dataset	
Normal	56.000	31,94%	37.000	44,94%
Analysis	2.000	1,14%	677	0,82%
Backdoor	1.746	1,00%	583	0,71%
DoS	12.264	6,99%	4.089	4,97%
Exploits	33.393	19,04%	11.132	13,52%
Fuzzers	18.184	10,37%	6.062	7,36%
Generic	40.000	22,81%	18.871	22,92%
Reconnaissance	10.491	5,98%	3.496	4,25%
Shellcode	1.133	0,65%	378	0,46%
Worms	130	0,07%	44	0,05%
Total	175.341	100,00%	82.332	100,00%

Intrusion Detection Evaluation Dataset (CIC-IDS2017)

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

SNo	S NoFeature Name	SNo	Feature Name	SNo	Feature Name	SNo	Feature Name
1	Flow ID	22	Flow Packets/s	43	Fwd Packets/s	64	Fwd Avg Packets/Bulk
2	Source IP	23	Flow IAT Mean	44	Bwd Packets/s	65	Fwd Avg Bulk Rate
3	Source Port	24	Flow IAT Std	45	Min Packet Length	66	Bwd Avg Bytes/Bulk
4	Destination IP	25	Flow IAT Max	46	Max Packet Length	67	Bwd Avg Packets/Bulk
5	Destination Port	26	Flow IAT Min	47	Packet Length Mean	68	Bwd Avg Bulk Rate
6	Protocol	27	Fwd IAT Total	48	Packet Length Std	69	Subflow Fwd Packets
7	Timestamp	28	Fwd IAT Mean	49	Packet Length Variance	70	Subflow Fwd Bytes
8	Flow Duration	29	Fwd IAT Std	50	FIN Flag Count	71	Subflow Bwd Packets
9	Total Fwd Packets	30	Fwd IAT Max	51	SYN Flag Count	72	Subflow Bwd Bytes
10	Total Backward Packets	31	Fwd IAT Min	52	RST Flag Count	73	Init_Win_bytes_forward
11	Total Length of Fwd Packets	32	Bwd IAT Total	53	PSH Flag Count	74	Init_Win_bytes_backward
12	Total Length of Bwd Packets	33	Bwd IAT Mean	54	ACK Flag Count	75	act_data_pkt_fwd
13	Fwd Packet Length Max	34	Bwd IAT Std	55	URG Flag Count	76	min_seg_size_forward
14	Fwd Packet Length Min	35	Bwd IAT Max	56	CWE Flag Count	77	Active Mean
15	Fwd Packet Length Mean	36	Bwd IAT Min	57	ECE Flag Count	78	Active Std
16	Fwd Packet Length Std	37	Fwd PSH Flags	58	Down/Up Ratio	79	Active Max
17	Bwd Packet Length Max	38	Bwd PSH Flags	59	Average Packet Size	80	Active Min
18	Bwd Packet Length Min	39	Fwd URG Flags	60	Avg Fwd Segment Size	81	Idle Mean
19	Bwd Packet Length Mean	40	Bwd URG Flags	61	Avg Bwd Segment Size	82	Idle Std
20	Bwd Packet Length Std	41	Fwd Header Length	62	Fwd Header Length	83	Idle Max
21	Flow Bytes/s	42	Bwd Header Length	63	Fwd Avg Bytes/Bulk	84	Idle Min

Other datasets

- Kyoto dataset
- CIC-IDS2017, CIC-DoS2017, CSE-CIC-IDS2018 and CIC-DDoS2019
- Korea University – Car hacking dataset
- UQ-IoT IDS dataset
- …

Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

- **I**nterpretable, **D**ependable and **S**ecure **Intrusion Detection System (IDS-IDS)**

1) Interpretable (eXplainable)
   - White/Black box -> Explanation in Natural language, clear boundary condition/visualization, etc.
   - To justify, control, improve and discover
2) Dependable
   - Safe and reliable against accidental/intentional events
3) Secure
   - against Poison, Extraction, **Evasion Attacks**

References:
- C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature machine intelligence, May 2019
- A. Dadai, M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access 2018
- H. Alemzadeh, Dependable AI Systems, IFIP WG 10.4., June 2017, available at: http://webhost.laas.fr/TSF/IFIPWG/Workshops&Meetings/72/ResearchReports/Alemzadeh-Dependable_AI_ML.pdf

Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey

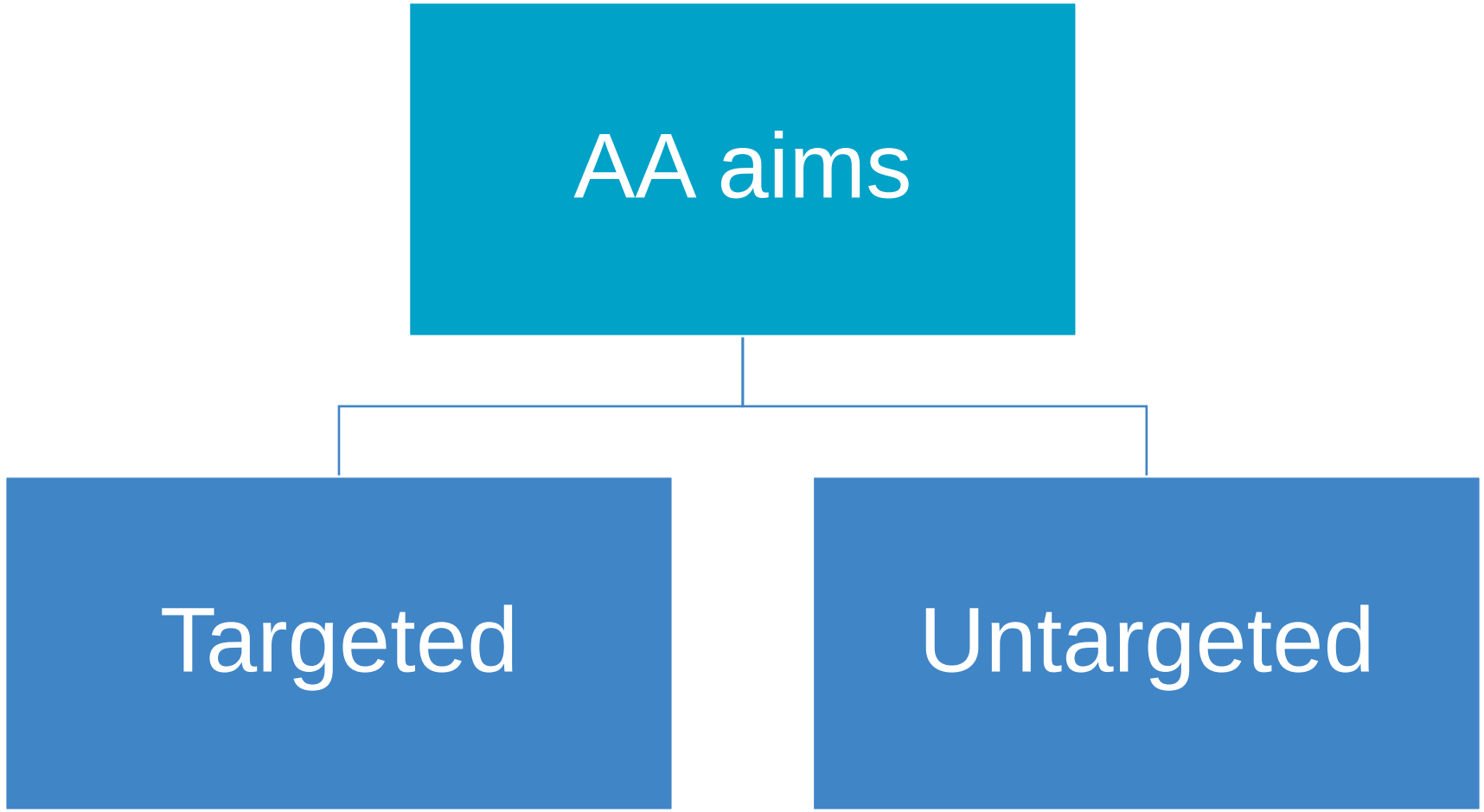
Ke He[1], Dongseong Kim[2], M. R. Asgahar[1,] J. Sun[1]

[1]University of Auckland, New Zealand

[2]The University of Queensland, Australia

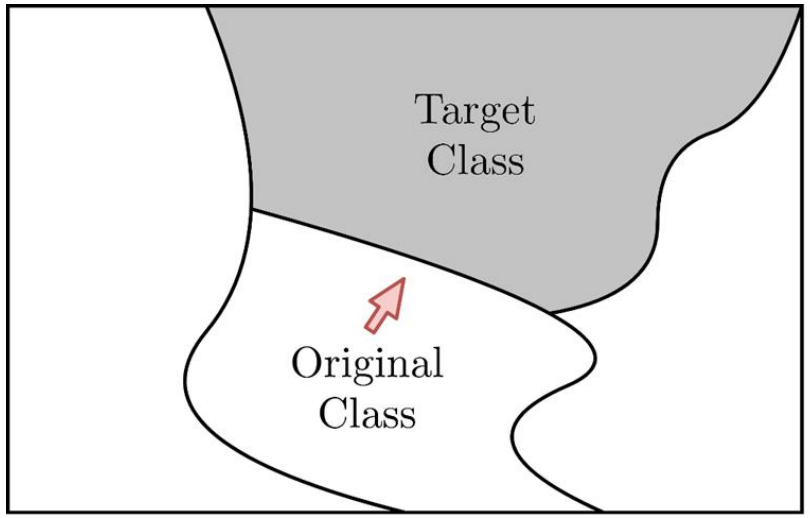
Published in the IEEE Communications Surveys and Tutorials (COMST) (impact factor: 25.249 (2021), #1 impact factor among all the IEEE journals)

Adversarial Attacks - aims

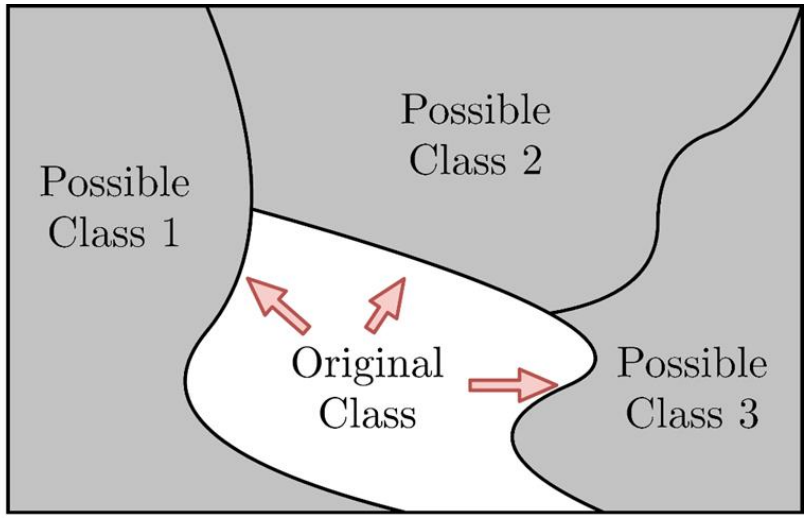


Adversarial Attacks

- Adversarial Attacks aims to fool a target ML/DL algorithm by adding imperceivable perturbations to the input so that its output is different from the original
- The attacks can be **targeted** or **untargeted**
  - Targeted attacks force the adversarial input to be classified as a predefined target class
  - Untargeted attacks force the adversarial input to be classified as any class that is not the original class



Targeted adversarial attack



Untargeted adversarial attack

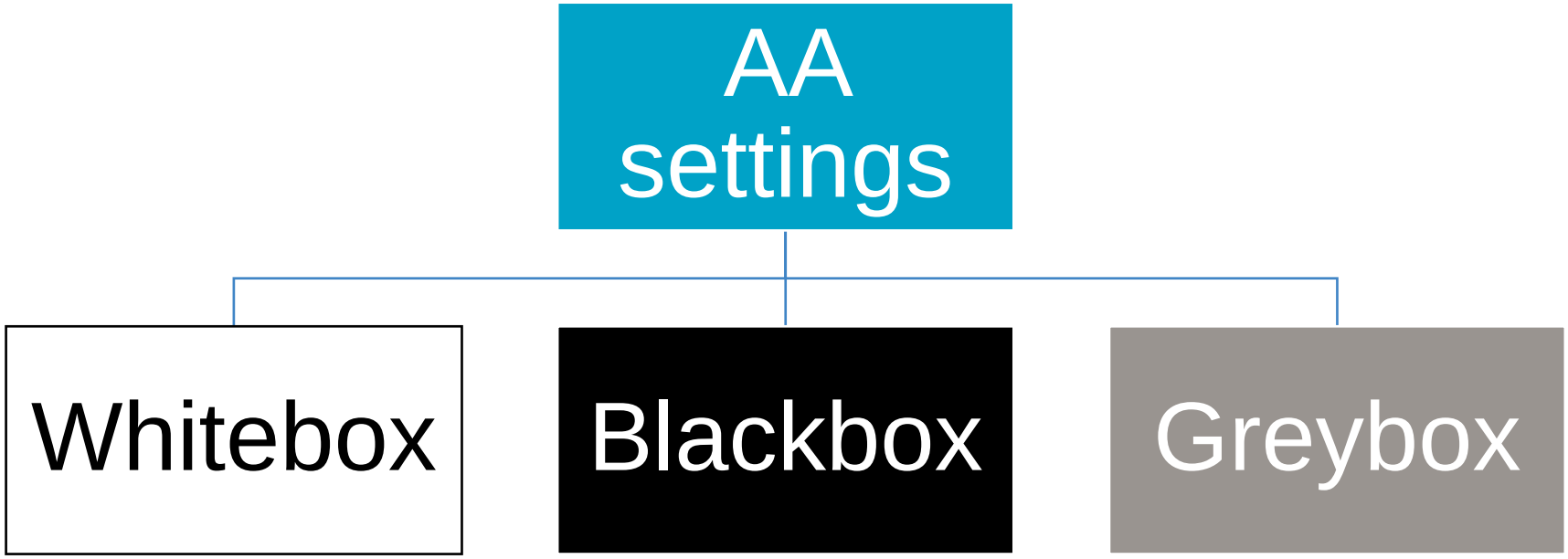
Generating Adversarial Attacks

- In general, adversarial attacks have to satisfy two constraints: Confidence and Similarity

- Confidence constraint ensures the input is classified incorrectly as the target class (targeted attacks) or not the original class (untargeted attacks).
  - For targeted attacks, the confidence of the target class is maximised
  - For untargeted attacks, the confidence of the original class is minimised

- Similarity constraint ensures the input is imperceivable by constraining the perturbation to be less than some threshold.

Adversarial Attacks (AA) settings



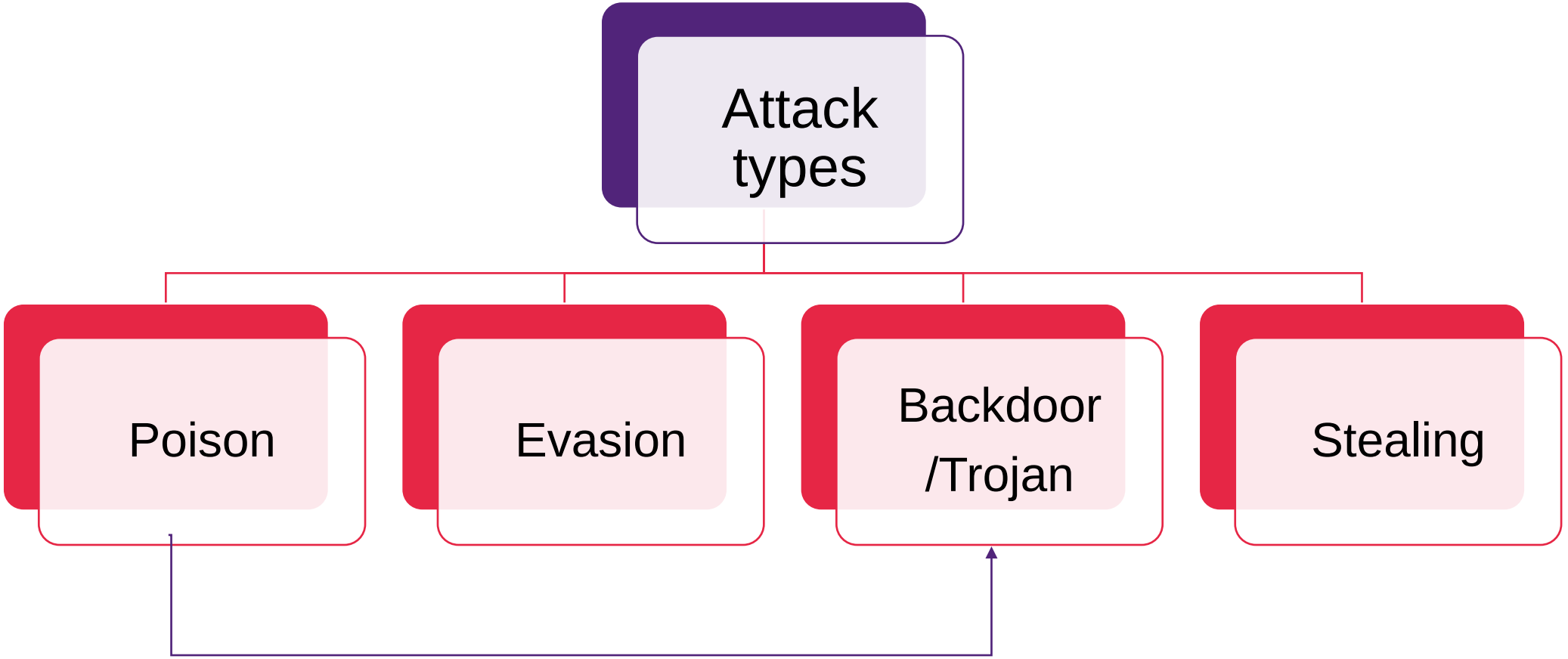
Generating Adversarial Attacks (cont.)

- The adversarial attacks can be generated mainly under threat models: white-box, black-box and grey-box

- White-Box (WB) assumes the attacker have completely knowledge of the target model.
  - Therefore, the search for adversarial input is done via gradient descent
  - Different white-box adversarial attacks place different emphasis on ensuring confidence and similarity.
  - In general, white-box attacks can be classified as Minimum Norm Attack, Maximum Allowable Attack, and Regularisation-based Attack

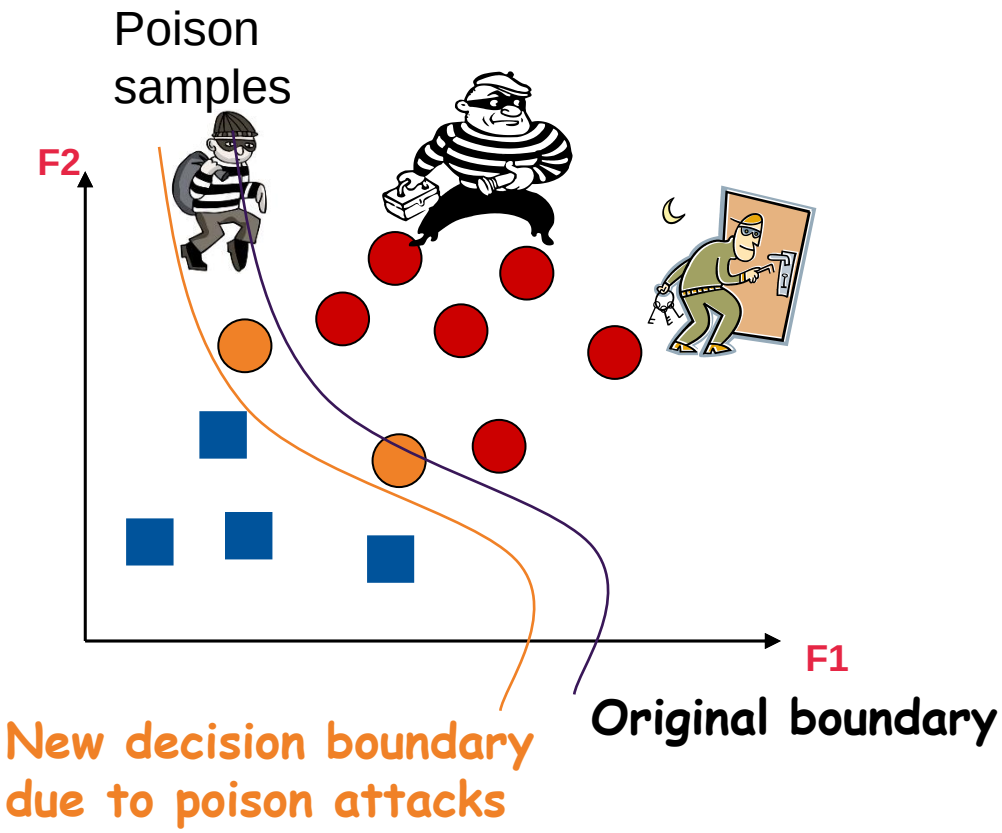
- Black-Box (BB) assumes the attacker have no knowledge of the target model.
  - The only output the attacker is able to obtain are the output decision or score. Moreover, the attacker can only have limited amount of queries to the target model.
  - Typical black-box attacks include Transfer-based Attack, Score-based Attack, and Decision-based Attack

- Grey-Box (GB) assumes the attacker have some knowledge of the target model.


Attack types
Poison
Evasion
Backdoor /Trojan
Stealing

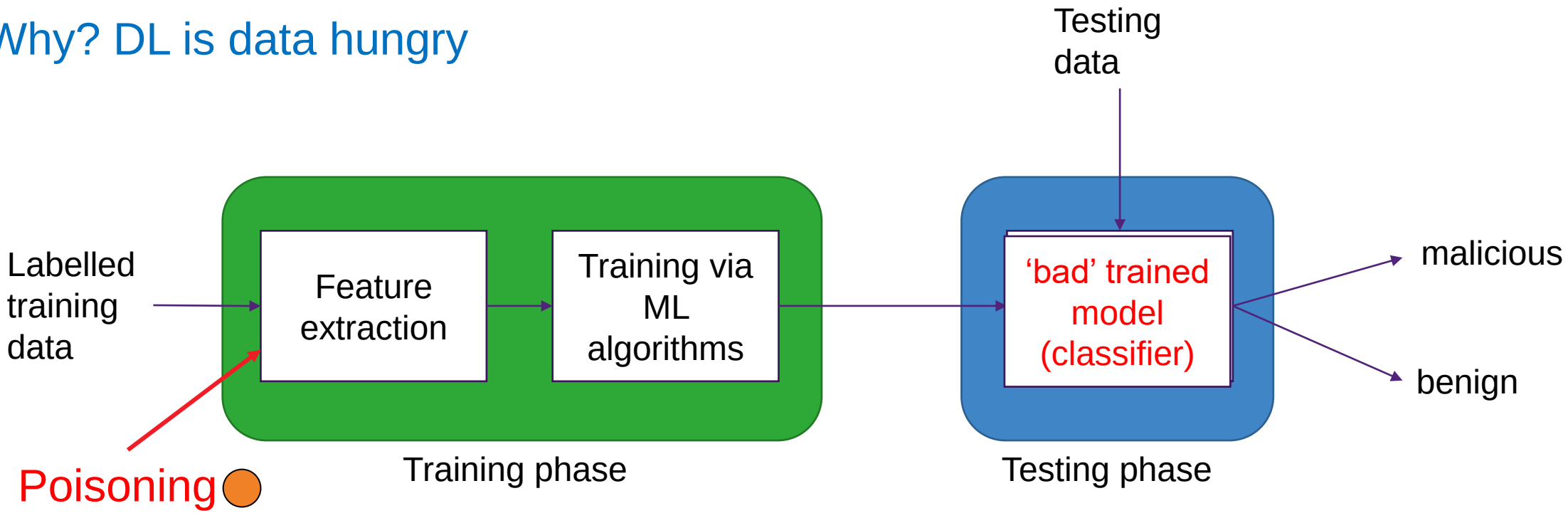
Poison attacks

- A ML/DL example:



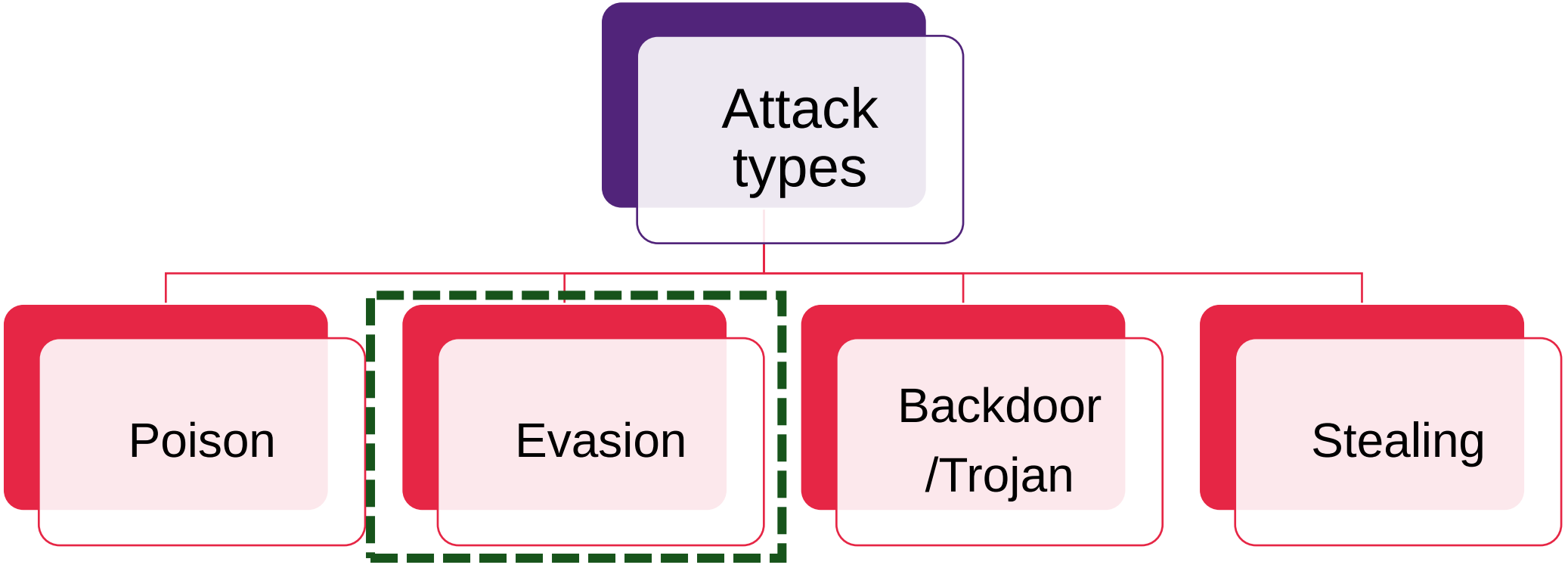
Poison attacks

Why? DL is data hungry

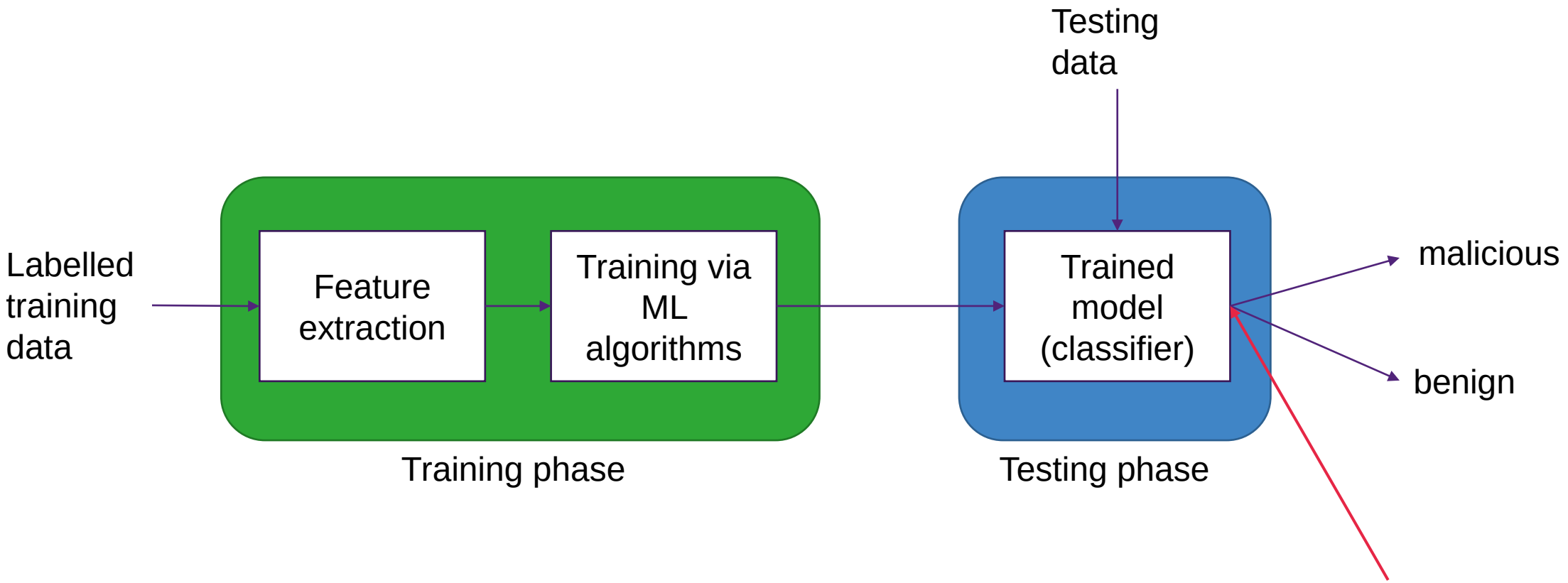


- An adversarial sample is an input crafted to cause ML/DL algorithms to misclassify. Adversarial samples are created at test time, after the ML/DL algorithm has been trained by the defender, and do not require any alteration of the training process

- David Evans, Classifier under Attack, USENIX Enigma 2017
- Papernot et al., The Limitations of Deep Learning in Adversarial Settings, Euro S&P 2016, https://arxiv.org/pdf/1511.07528.pdf


Attack types
Poison
Evasion
Backdoor /Trojan
Stealing

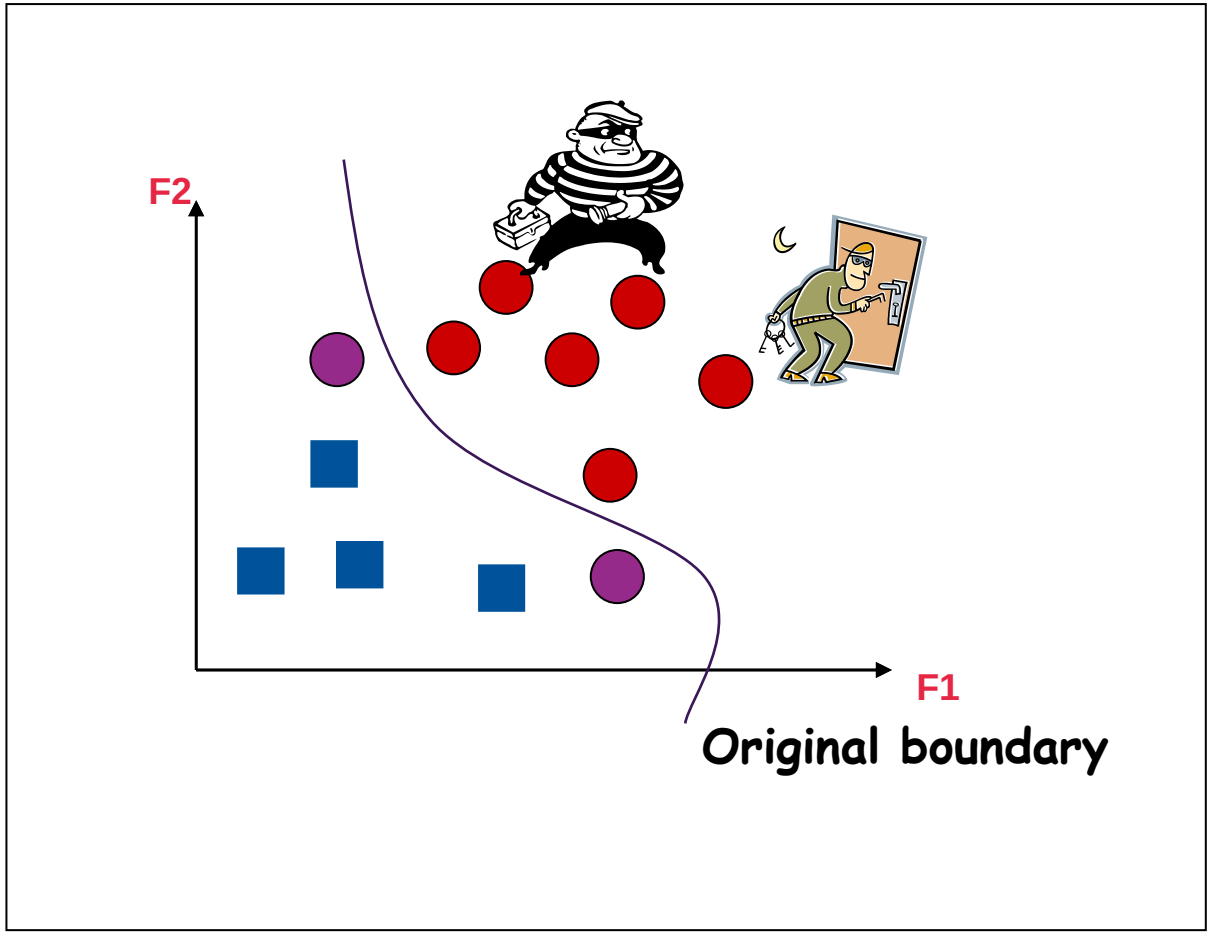
Evasion attacks



- Find samples which a trained model can misclassify.

Evasion attacks – evade trained model(s); e.g., evade a classifier

Evasion attacks



A summary of DL-based NIDS architectures.

TABLE II: A summary of DL-based NIDS architectures from 2017 to 2020.

Solution	Year	Target Network	Evaluation Datasets	Feature Extraction	Feature Reduction	Detection Paradigms	Detection Algorithms	Application	Main Metrics
Lopez *et al.* [39]	2017	IoT	NSL-KDD	Flow-based	None	Classification	CVAE	Offline	Accuracy, F1
Yousefi *et al.* [40]	2017	General	NSL-KDD	Flow-based	AE	Classification	SVM XGBoost	Offline	Accuracy
Yin *et al.* [41]	2017	General	NSL-KDD	Flow-based	None	Classification	RNN	Offline	Accuracy, FPR, Time
Thing [42]	2017	Wireless	Own Dataset	Packet-based	AE	Outlier Detection Classification	Softmax Regression	Real-time	Accuracy
Jia *et al.* [43]	2018	General	NSL-KDD KDD Cup'99	Flow-based	None	Classification	DNN	Offline	F1, FPR, FNR
Shone *et al.* [44]	2018	General	NSL-KDD KDD Cup'99	Flow-based	SNDAE	Classification	Random Forest	Offline	Accuracy, F1, FPR, Time
Yan *et al.* [45]	2018	General	NSL-KDD	Flow-based	SSAE	Classification	SVM	Offline	Accuracy, FPR, TPR, Time
Naseer *et al.* [46]	2018	General	NSL-KDD	Flow-based	None	Classification	CNN RNN AE	Offline	Accuracy, AUC, Time

A summary of DL-based NIDS architectures (continued)

Mirsky *et al.* [47]	2018	IoT	Kitsune	Packet-based	None	Outlier Detection	AE Ensemble	Real-time	AUC, EER, Time
Diro *et al.* [48]	2018	IoT	NSL-KDD	Flow-based	None	Classification	DNN	Offline	Accuracy, F1, TPR, FPR
Otoum *et al.* [49]	2019	IoT	NSL-KDD	Flow-based	SMO	Classification	SDPN	Offline	Accuracy, F1
Xiao *et al.* [50]	2019	General	KDD Cup'99	Flow-based	PCA AE	Classification	CNN	Offline	Accuracy, TPR, FPR, Time
Zhang *et al.* [51]	2019	General	UNSW-NB15 CIC-IDS2017	Packet-based	None	Classification	CNN & caXGBoost	Real-time	Accuracy, TPR, FPR, Time
Vinayakumar *et al.* [52]	2019	General	NSL-KDD KDD Cup'99 UNSW-NB15 CIC-IDS2017 WSN-DS	Flow-based	None	Classification	DNN	Offline	Accuracy, F1
Roopak *et al.* [53]	2019	IoT	CIC-IDS2017	Flow-based	None	Classification	CNN & LSTM	Offline	Precision, Recall, Accuracy
Nguyen *et al.* [54]	2019	General	UGR'16	Flow-based	None	Outlier Detection	VAE	Offline	AUC
Sun *et al.* [55]	2020	General	CIC-IDS2017	Flow-based	None	Classification	CNN & LSTM	Offline	Accuracy, F1

Adversarial Attacks in NIDS

- Adversarial attacks have been mostly studied in Computer Vision (CV). However, adversarial attacks can cause much more significant damage in security-sensitive domains such as Network Intrusion Detection System (NIDS)

- A successful adversarial attack can cause malicious packets bypass ML/DL based NIDS, which potentially violates the confidentiality, integrity, and availability of the network.

Adversarial Attacks in NIDS (cont.)

- Adversarial attacks in NIDS have fundamental differences compared to CV
- Due to the large amounts of network packets being transmitted, the NIDS extracts numerous features from the packets and classifies the features. Therefore, simply altering the features does not create transmittable adversarial packets.
- Network features are highly correlate. Therefore, any perturbation to the features must convey the correlations. However, the exact correlation between the features are mostly unknown.
- Network features are also sequentially related. Hence, modification of a single packet can affect the features of the subsequent packets.

Adversarial Attacks – types

- Feature-Level Attacks (FLA)
  - directly perturb the **input network traffic features**
  - Although these studies show WB and BB attacks with feature-level perturbations are effective, they have a common limitation: lack of practicality.
- Generative Feature-Level Attacks (GFLA)
  - utilize generative algorithms such as Generative Adversarial Networks (GAN) to learn the inherent structure within network traffic features and enforce the features to look realistic.
  - capture the hidden interdependencies between network features and constrain the feature-level perturbations to be more realistic.
  - However, despite realistic adversarial network features, there is still a significant gap in transforming the network features **into replayable packets**.
- Packet-Level Attacks (PLA)
  - **manipulate the network packets directly** rather than network features.
  - mark a significant improvement in practical adversarial attacks evading AI based NIDS, as they can generate replayable packets in the network.
  - However, existing adversarial attacks lack a comprehensive evaluation of **the maliciousness** of the adversarial examples.

Adversarial Attacks – Threat models

- Goals (constraints)
  - Bounded by l_p ball, where the network features are freely perturbed with an l_p ball.
  - Bounded by the generative model, where the network features are constrained by generative models such as GAN to be similar to realistic network features
  - Bounded by the packet obfuscation, where packet-level perturbations are bounded by pre-defined mutation operations.
- Knowledge
  - BB: if the attacker has zero knowledge of the classifier and its auxiliary components.
  - GB: if the attacker has zero knowledge of the classifier but has some knowledge of the pre-processing functions.
  - WB: if the attacker has complete knowledge of the classifier and auxiliary components.
- Capability
  - None, if the attacker cannot obtain the training dataset, or it is not necessary (e.g., the attacker has WB knowledge)
  - Passive, if the attacker can capture both benign and malicious traffic.
- Target models
  - ML, DL, …

TABLE III: Comparison of adversarial attacks designed to bypass NIDS.

Author	Year	Type	Constraints	Knowledge	Capability	Target Model	Datasets	Metrics	Main Limitations
Rigaki *et al.* [67]	2017	FLA	ℓ_p ball	WB	None	Shallow ML	NSL-KDD	Accuracy, F1, AUC	Outdated dataset
Homoliak *et al.* [104]	2018	PLA	Packet obfuscation	GB	None	Shallow ML	ASNM-NPBO	F1	Vigorous Trial and Error
Lin *et al.* [105]	2018	GFLA	Generative model	GB	Passive	Shallow ML	NSL-KDD	Accuracy	Outdated dataset
Wang [68]	2018	FLA	ℓ_p ball	WB	None	MLP	NSL-KDD	Accuracy, F1, ROC	Outdated dataset
Yang *et al.* [69]	2018	FLA	ℓ_p ball	GB	Passive	DNN	NSL-KDD	F1	Outdated dataset
Clements *et al.* [106]	2019	FLA	ℓ_p ball	WB	None	Kitsune	Kitsune	Accuracy, ℓ_p distance	Unrealistic adversarial features
Hashemi *et al.* [107]	2019	PLA	Packet obfuscation	WB	None	Kitsune, DAGMM, BiGAN	CIC-IDS2018	TPR, FPR	Vigorous Trial and Error
Ibitoye *et al.* [108]	2019	FLA	ℓ_p ball	WB	None	FNN, SNN	BoT_IoT	Accuracy, F1, Cohen's Kappa, MCC	Unrealistic adversarial features
Kuppa *et al.* [109]	2019	PLA	MAA+Packet obfuscation	GB	Passive	AGMM, AE, AnoGAN, ALAD, DSVDD, shallow ML	CIC-IDS2018	F1	Lack of maliciousness evaluation
Alhajjar *et al.* [110]	2020	GFLA	Generative model	GB	None	Shallow ML	NSL-KDD, UNSW-NB15	Accuracy, FNR	Lack of replayability
Han *et al.* [70]	2020	PLA	Generative model + Packet obfuscation	BB/GB	Passive	Kitsune	Kitsune	FNR, ℓ_2 Distance	Lack of maliciousness evaluation
Chen *et al.* [111]	2021	GFLA	Generative model + Domain constraints	BB	Passive	Shallow ML	CTU-13	MAPE, FNR	Lack of replayability
Sharon *et al.* [112]	2021	PLA	LSTM	BB	Passive	Kitsune, AE, IF	Kitsune, CIC-IDS2017	TPR	Only considers time delay
Sheatsley *et al.* [113]	2022	FLA	Extracted constraints	WB	None	Shallow ML	NSL-KDD, UNSW-NB15	Accuracy	Lack of replayability
Zolbayar *et al.* [114]	2022	GFLA	Generative model + Domain constraints	WB/GB/BB	Passive	DNN, MLP	NSL-KDD, CIC-IDS2018	FNR	Lack of replayability

Type:
- Feature-Level Attacks (FLA)
- Generative Feature-Level Attacks (GFLA)
- Packet-Level Attacks (PLA)

Knowledge:
- Whitebox (WB)
- Greybox (GB)
- Blackbox (BB)

Capability:
- Passive
- None

Defenses against Adversarial attacks

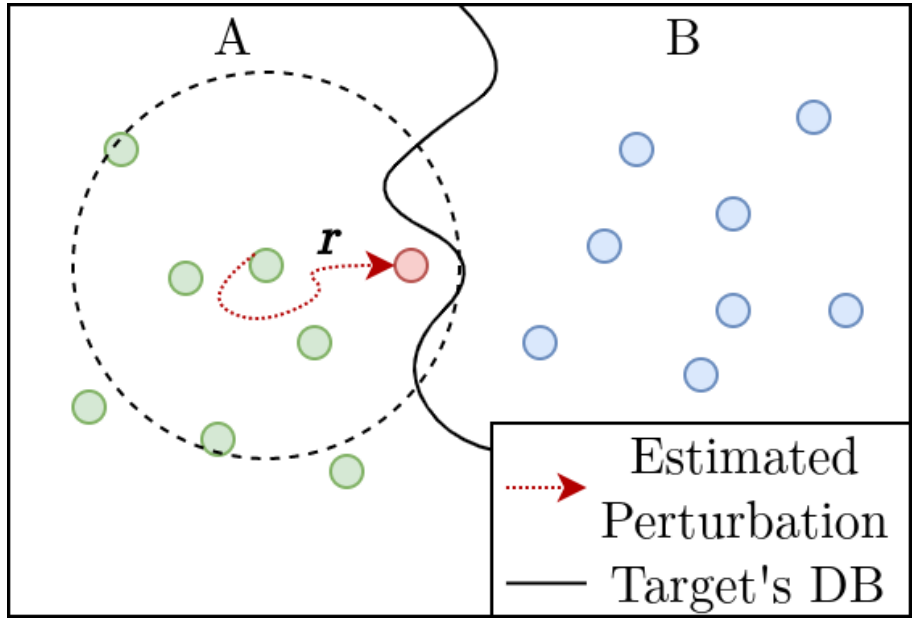
- have been encouraged to increase the robustness of the model so that they can correctly classify the adversarial input/attack(s).
- are created by studying the formulation and characteristics of the adversarial examples and designed to exploit its weaknesses.
- Common defenses include
  - **Gradient Masking** - prevents the gradient of the model to be exposed by the attacker, even under white-box setting
  - **Adversarial Detection** - adversarial examples possess unique characteristics that can be used to distinguish between adversarial and clean input
  - **Robustness Optimization** - increase the robustness of the model directly so it is able to correctly classify the adversarial input/attacks

TABLE IV: Comparison of various defence techniques against adversarial attacks and their applicability in the NIDS domain.

Type	Methods	Brief Description	Defence Strength	Applicability	Section
Parameter Protection	Gradient Masking	Hides the gradient of the model from attackers to prevent WB attacks.	Weak defence. BB attacks do not require gradient and can easily bypass gradient masking.	Applicable since gradient masking is domain agnostic.	VI-A
Adversarial Detection	Secondary Classifier	Uses another neural network to classify clean and adversarial examples	Weak defence. The attack can be formulated to simultaneously bypass both detectors under the limited-knowledge and perfect-knowledge threat model.	Not applicable. Adversarial examples are not available at training time.	VI-B1
	Projection-Based	Projects input into low-dimensional space to detect adversarial examples.	Weak defence. Characteristics in low-dimensional space are specific to datasets.	Applicable but not generalisable.	VI-B2
	Statistics-Based	Finds distributional differences between clean and adversarial examples.	Weak defence. Statistical characteristics are specific to datasets.	Applicable but not generalisable.	VI-B3
	Mutation-Based	Mutates the decision boundary randomly to detect adversarial example.	Strong defence. The adversarial attack has to be universally adversarial to bypass this defence.	Applicable.	VI-B4
Robustness Optimisation	Input Preprocessing	Filter or transform the input to remove adversarial perturbation.	Weak defence. The transformations can be modelled and bypassed with EOT.	May be applicable, but the transformations may remove malicious traits.	VI-C1
	Adversarial Training	Trains the model with a correctly labelled adversarial example.	Strong defence. Adversarial training finetunes the decision boundary of the classifier.	Not applicable. Adversarial examples are not available at training.	VI-C2
	Post-Training	Modifies the structure of the model after training.	Weak defence. Multiple stronger attacks can bypass post-training defences.	Applicable but weak defence.	VI-C3

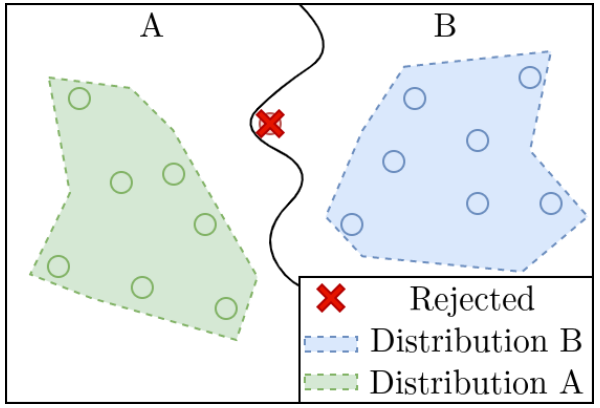
Gradient Masking

- White-box adversarial attacks requires the gradient to be known in order to conduct gradient descent.
- Gradient masking aims to hide the gradient of the white-box model from the attacker, so that gradient descent cannot find a valid perturbation
- Gradient masking can be done via
  - **Shattered Gradients** - the gradient is non-differentiable, numerically unstable, non-existent or incorrect.
  - **Stochastic Gradients** - the gradient is randomised
  - **Vanishing & Exploding Gradients** - the gradient vanishes to 0 or explodes to infinity
- Gradient masking is limited against black-box attacks since black-box attacks does not require any gradient information

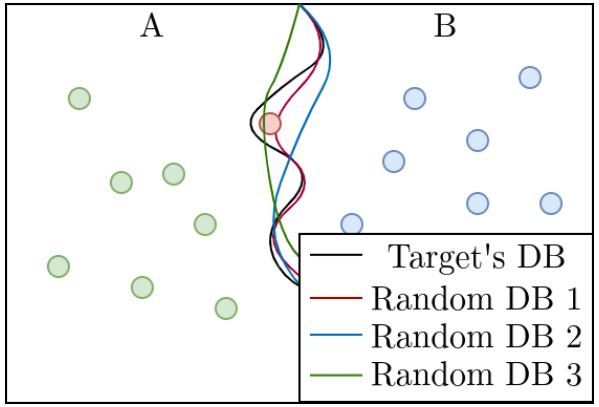


Adversarial Detection

- aims to detect and reject the adversarial examples.
- is often implemented in a plug-and-play manner and does not require any modification to the model
- Adversarial Detection methods include:
  - **Distributional-based** - adversarial examples are artificially created and does not naturally occur in the input space. Therefore, it possess unique characteristics that are distinguishable from clean examples
  - **Mutation-based** - the similarity constraint makes the adversarial examples lie close to the decision boundary, mutating the decision boundary will cause the output of the adversarial example to be drastically different



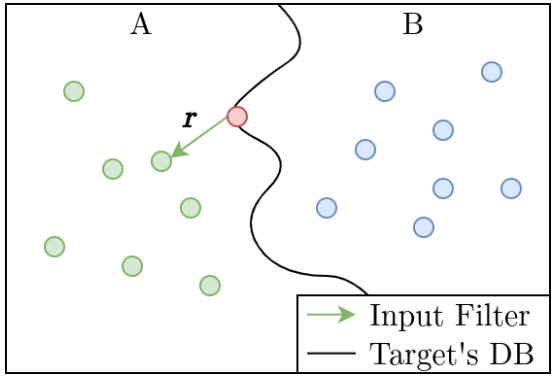
Distribution-based



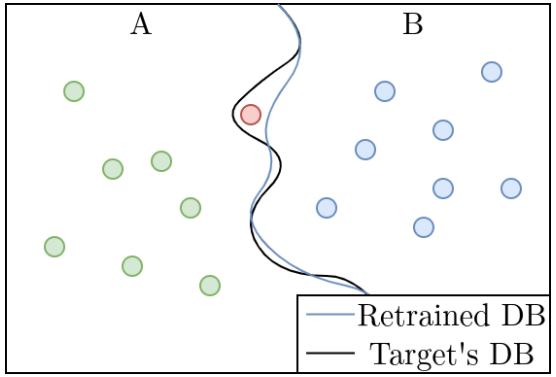
Mutation-based

Robustness Optimisation

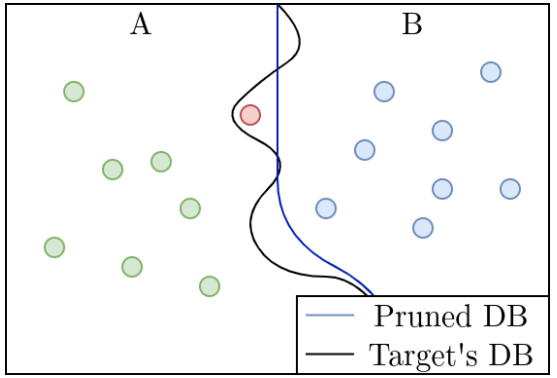
- aims to ensure the adversarial examples is correctly classified by the model
- Common methods include
  - **Input pre-processing** - adversarial perturbations are small, therefore applying filters can remove the perturbations
  - **Adversarial training** - train the model with correctly labelled adversarial examples so that it learns the adversarial example
  - **Post-training modifications** - the model's decision boundary is smoothed so that it can generalise better for data outside of the training dataset



Input Preprocessing



Adversarial training



Post-training modification

Outline

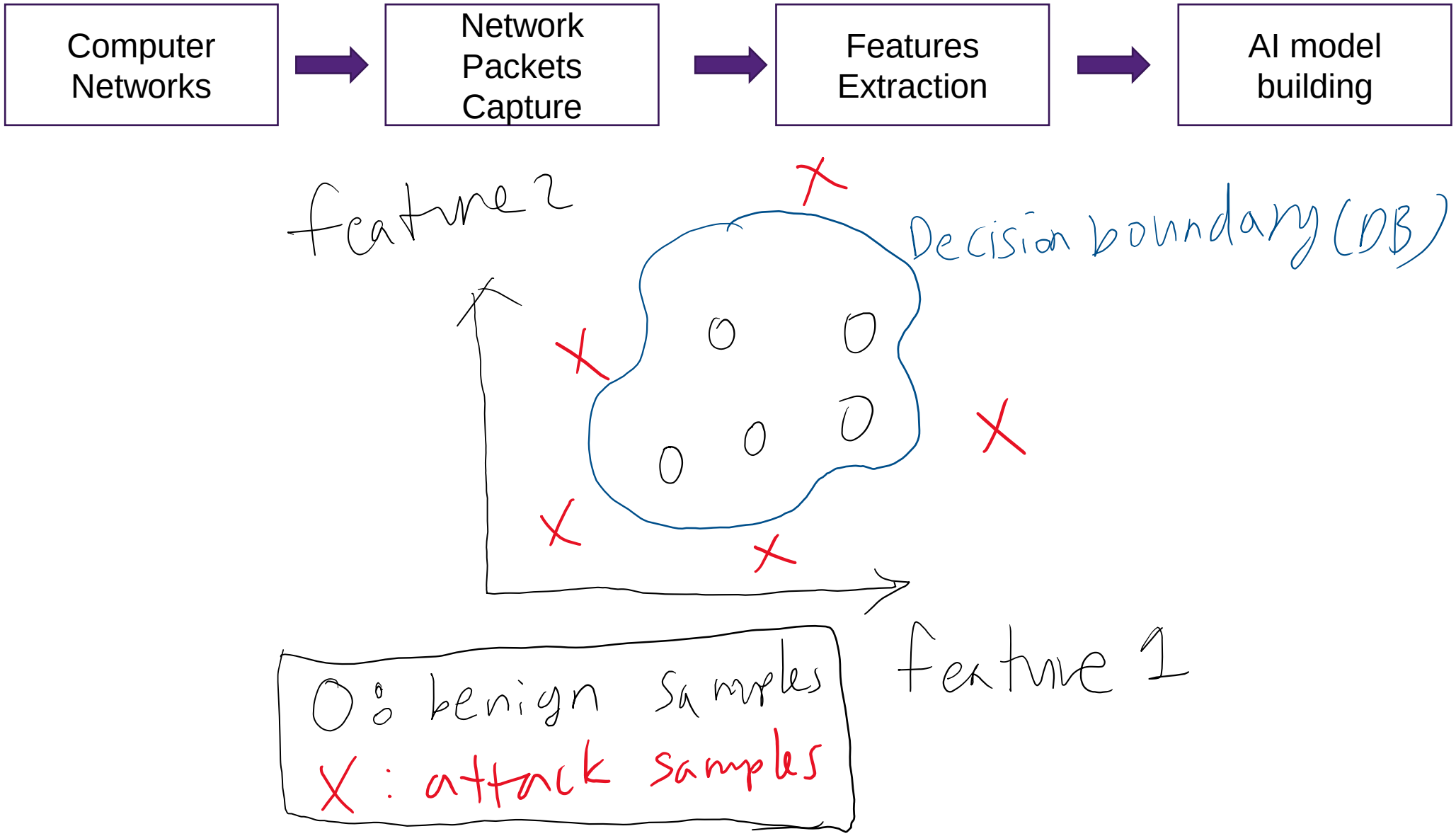
- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

Liuer Mihou: A Practical Framework for Generating and Evaluating Grey-box Adversarial Attacks against NIDS

Ke He, Dongseong Kim, Jing Sun, Jeong Do Yoo, Young Hun Lee, Huy Kang Kim

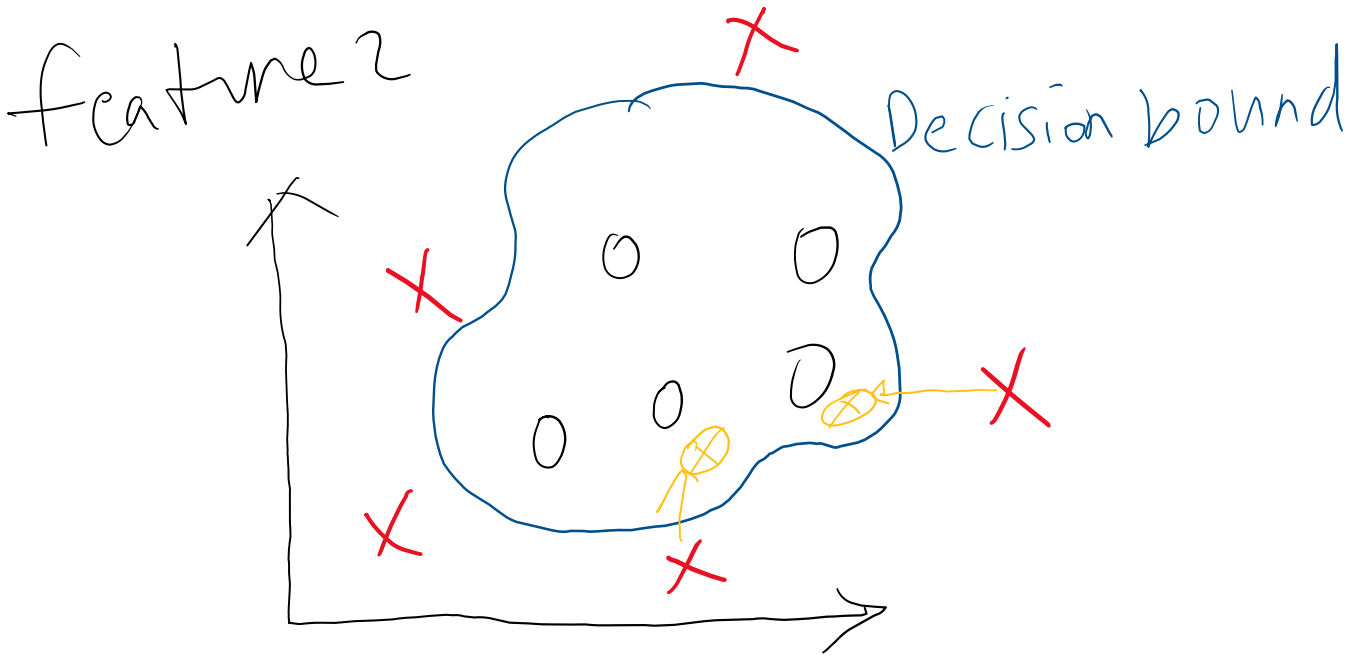
CoRR abs/2204.06113 (2022) / currently in IEEE TDSC (under review)

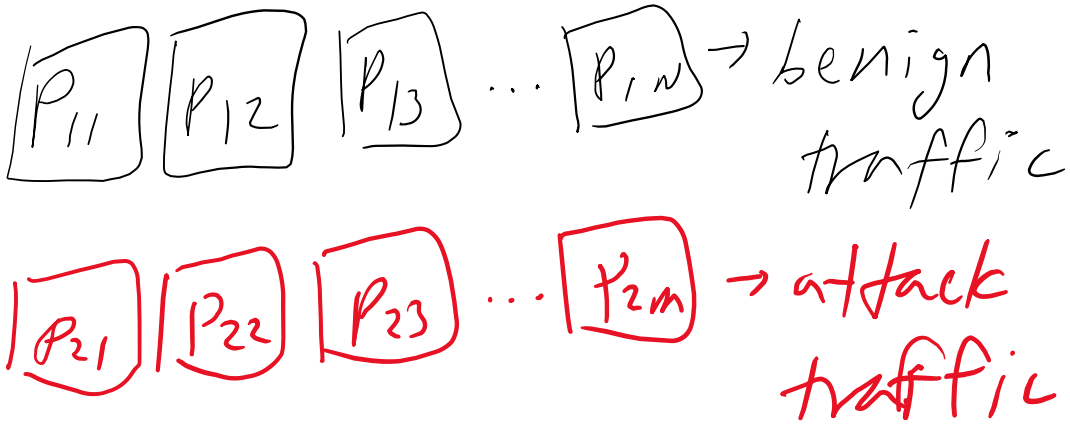
Answer for Q1

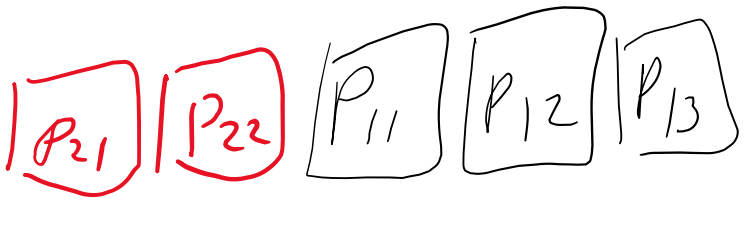


Can we fool an AI model?

Partial answer for Q2



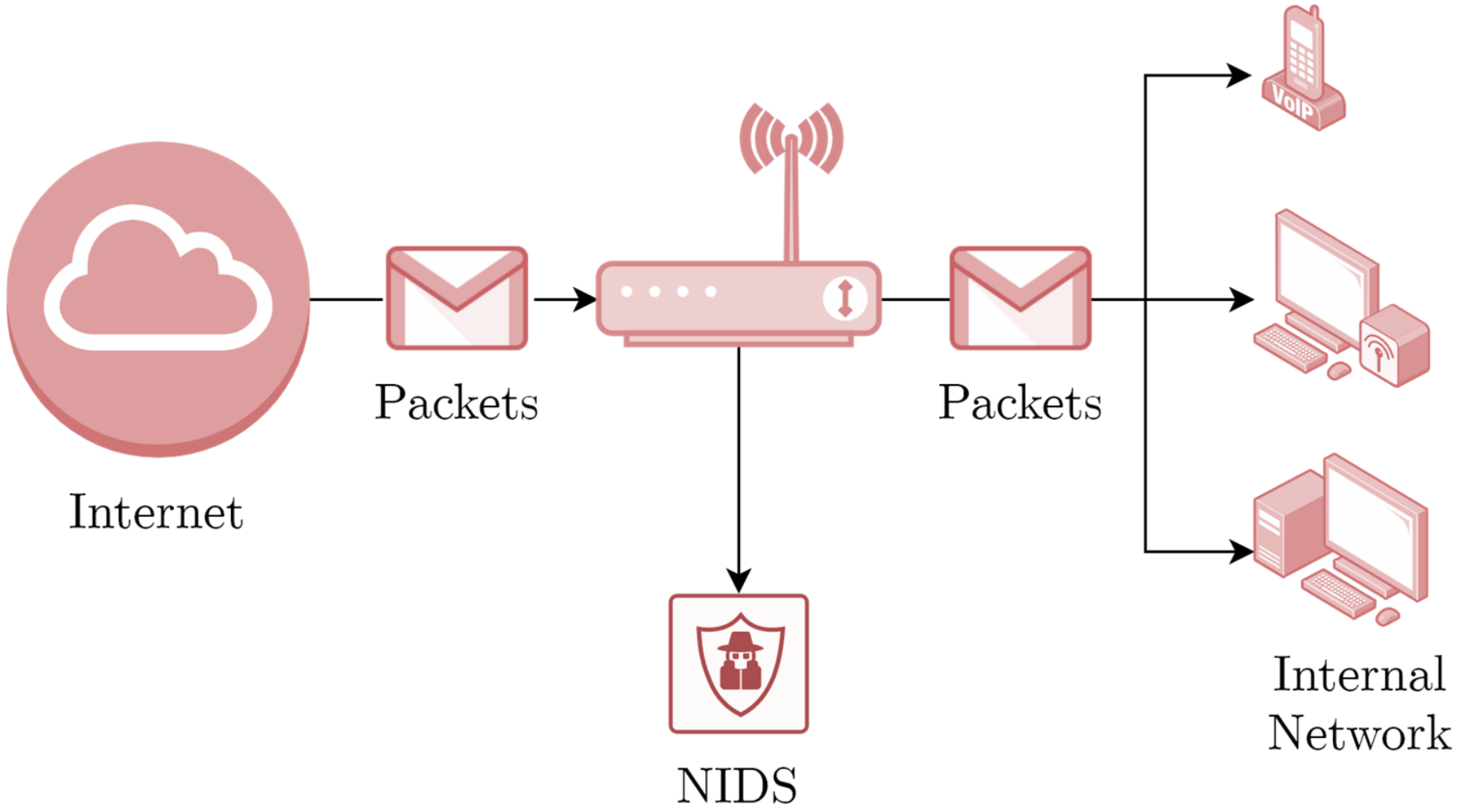




Introduction

- Deep learning have been used extensively in many domains
- Network Intrusion Detection Systems (NIDS) is not an exception
- Deep learning algorithms are known to be vulnerable to adversarial examples
- Unlike Computer Vision (CV), adversarial examples in NIDS can have a serious impact
  - But most work on adversarial examples are done in CV, not many in NIDS
  - Those that have been done in NIDS have done it rather poorly
- We would like to bridge this gap

Background - NIDS



Background - NIDS

- Extracts aggregate features from the packets
  - Statistics about packet sizes of a connection or in the last few seconds
  - Inter arrival times of a connection
  - Flags
- Due to the large volume of packets and the use of encryption, the features does not require examining the payload
- The extracted features are used by detection algorithm to be classified as benign or malicious

TABLE II: The statistics (features) extracted from each time window λ when a packet arrives.

The packet's...	Statistics	Aggregated by	# Features	Description of the Statistics
...size	μ_i, σ_i	SrcMAC-IP, SrcIP, Channel, Socket	8	*Bandwidth of the outbound traffic*
...size	$\|\|S_i, S_j\|\|, R_{S_i,S_j}, Cov_{S_i,S_j}, P_{S_i,S_j}$	Channel, Socket	8	*Bandwidth of the outbound and inbound traffic together*
...count	w_i	SrcMAC-IP, SrcIP, Channel, Socket	4	*Packet rate of the outbound traffic*
...jitter	w_i, μ_i, σ_i	Channel	3	*Inter-packet delays of the outbound traffic*

Table source: Mirsky, Yisroel, et al. "Kitsune: an ensemble of autoencoders for online network intrusion detection." *NDSS* (2018).

Background - Adversarial Examples on NIDS

- Previous works have mostly used adversarial attacks in CV and applied it directly to modify traffic features
  - Does not achieve much, since the features cannot be used to do anything
- Some have modified packets
  - A little bit more useful, but they often take random guesses and trial and error
  - No evaluation of maliciousness of adversarial example
- We want to formalize adversarial attacks on NIDS and generate realistic adversarial examples (packets) that are malicious

Contributions

- We design a novel and practical adversarial attack tailored to attack anomaly-based NIDS called Liuer Mihou (LM) attack, and provide source code for download.
  - https://github.com/XXXX-5/automatic-waddle
- We conduct a comprehensive evaluation of LM attack in a real IoT testbed against five ML-based anomaly detection algorithms (SOM, RRCF, LOF, OCSVM, and FROCC), and the state-of-the-art DL based IoT NIDS, Kitsune.
- We demonstrate the strength of LM attack by assessing our attacks on Kitsune with adversarial detection defences (such as Feature Squeezing and Mag-Net).
- We show empirical results and findings of our experiments, which provide insights for future adversarial attacks against NIDS.

Threat Model

- Our goal
  - Modify an existing, detectable sequence of malicious packets to evade detection
  - Preserve the malicious behaviour of the packets to some extent
- Focus on IoT networks that is much simpler
  - Small network size, devices do not have much computation power, and is relatively more periodic
- We know the NIDS uses outlier detection algorithm(s) to classify packets
  - More realistic compared to traffic classification
- We know the features extracted
  - Reasonable assumption, since there are only a certain number of features that can be extracted
- We can sniff the traffic in the network
  - IoT networks are often connected wirelessly can sniffing packets is rather easy.

Liuer Mihou (LM) attack framework - introduction

- a practical adversarial generation algorithm tailored specifically for NIDS that operates iteratively on each packet of the malicious traffic.
- For each packet, the surrogate NIDS first classifies the packet.
- If the surrogate classifies the packet as malicious, it is likely to be classified as malicious by the target model.
- For each malicious packet, LM first searches for an optimal set of mutation operations on the packet that minimizes the anomaly score produced by the surrogate NIDS.
- Next, the optimal mutation operations are applied to the malicious packet, transforming it into adversarial packets containing the modified malicious packet and some redundant packets as byproducts.
- Finally, the adversarial packets are written to the output file in place.

Practical Evasion Attack Generation for NIDS

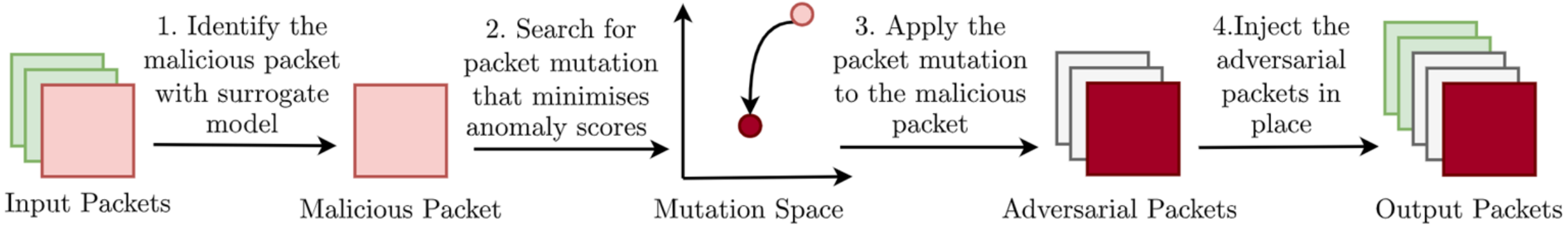
Train a surrogate detection algorithm given the features and benign traffic

Iteratively classify each malicious packet with the surrogate model

If it is classified as malicious, search for a set of mutation operations that minimises the anomaly score

Inject the adversarial packets in place and evaluate

Liuer Mihou attack framework



- Train a surrogate detection algorithm given the features and benign traffic
- Iteratively classify each malicious packet with the surrogate model
- If it is classified as malicious, search for a set of mutation operations that minimises the anomaly score
  - Delay the packet by some time
  - Inject redundant packets in front of the malicious packet
- With the mutation operations, create a low-dimensional space where each dimension represents a mutation operation
- The search is done via heuristics search algorithms, in our case, a hybrid between PSO and DE
  - Particle Swarm Optimization (PSO) and the Differential Evolution (DE) algorithms
- Inject the adversarial packets in place

Mutation Operation and Objective Function

- A set of mutation operations are can be applied to a malicious packet to change the extracted features of the packet with minimal change of the content.
  - Packet Delay: Delay the arrival time of a packet, which changes the inter-arrival time distribution.
  - Packet Injection: Inject redundant packets before a packet in the same connection, which changes packet-size distribution without affecting the original payload.
- The objective function formulated as the optimisation problem, done for each malicious packet (p_i):

$$\min_{m \in \mathcal{M}} \max\{g'(\varphi(p')) : p' \in m(p_i)\}$$

  - where *M* is the space of all possible mutations, m is a specific mutation operation, p_i is the i^{th} malicious packet, p' is the set of adversarial packets after mutation, g' gives the anomaly score based on the input feature.
- The adversarial packets contain the modified malicious packet and several redundant packets.
- The aim is to reduce the maximum anomaly score of the adversarial packets so not to introduce more malicious packets.
- In case there are still malicious packets above the surrogate threshold, we can recursively run LM on the output adversarial packets until it reduces all packets below the surrogate threshold.

Packet Vectorisation

- To efficiently search for the mutation operations, representation of mutation operations into low-dimensional vectors is abstracted in the mutation space (Ψ, psi).
- With the mutation operations, defined Ψ as a two-dimensional space with (t_m, n_c):
  - Modified arrival time of the packet (t_m), which represents packet delay.
  - Number of redundant packets inserted before the packet (n_c), which represents packet injection.
  - For example, (0.4, 4): delay the malicious packet by 0.4 seconds and place four redundant packets before the malicious packet.
- For packet injection, the three methods of assigning arrival time and payload size of the redundant packets.
  - Random Assignment (RA). Seeded Assignment (SA). Uniform Assignment (UA)

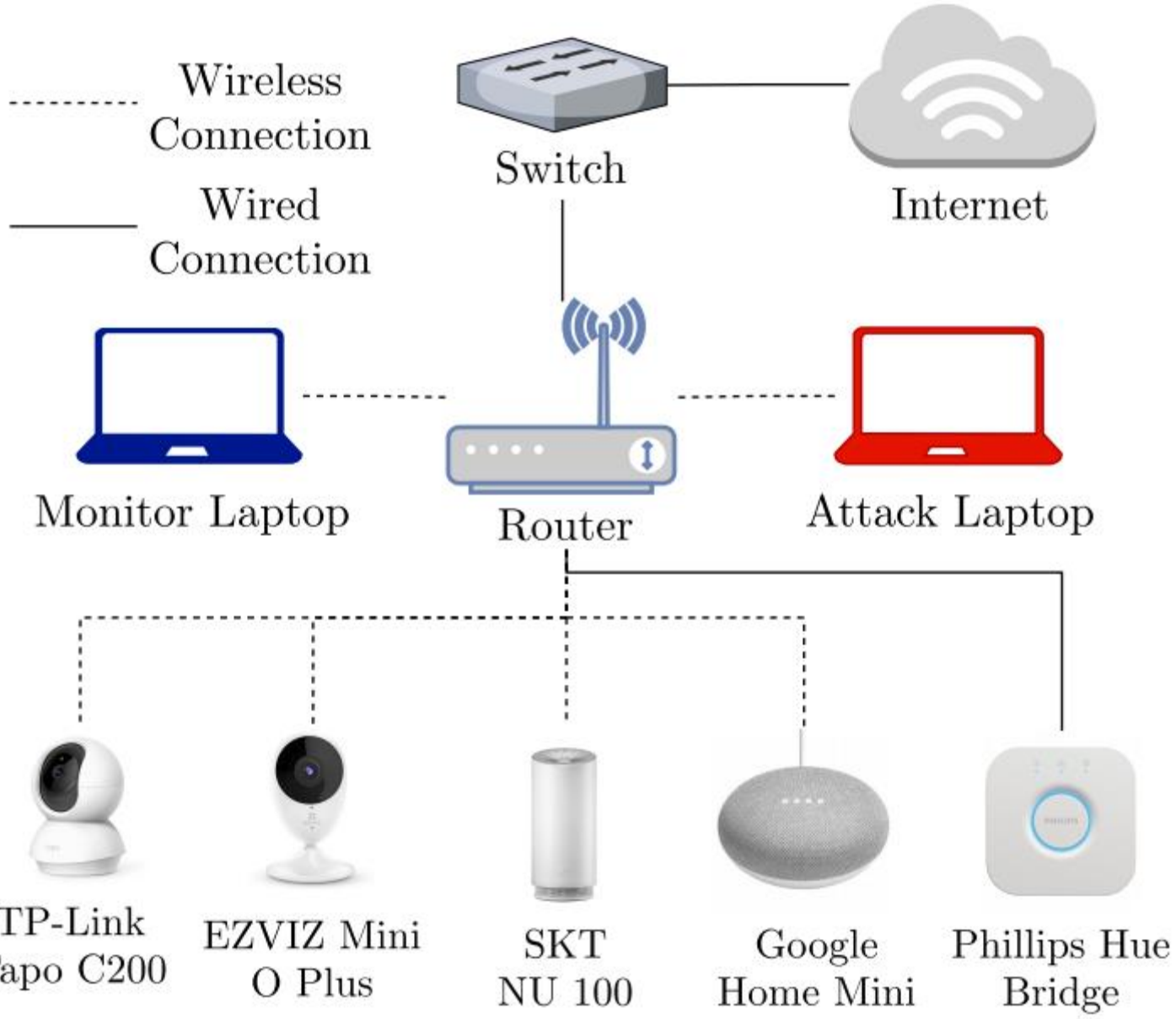
Injection methods

- Three methods of assigning arrival time and payload size of the redundant packets.
  - Random Assignment (RA). randomly assigns the arrival time and payload size of each redundant packet. We have found this method causes the cost function to be non-deterministic because the arrival time and payload size of the redundant packets at the same position are different in each iteration
  - Seeded Assignment (SA). SA seeds the random number generator with the value of n_c before generating the payload sizes for each redundant packet.
  - Uniform Assignment (UA). A new dimension is introduced in the mutation space, sc , which governs the payload size of all redundant packets. As a result, the cost value is deterministic and piece-wise constant around each integer value of n_c
- Selected UA for creating redundant packets, which suggest that uniform payload sizes are also characteristic of benign traffic in the dataset.

Search Algorithms

- The typical approach to solve optimization problems involving non-differentiable and non-invertible functions is to use meta-heuristic algorithms.
- Meta-heuristic algorithms often have a master strategy that iteratively generates solutions, and the optimal solution is discovered by continuously evolving the solutions according to the fitness function.
- Liuer Mihou utilizes a hybrid heuristic search algorithm that combines PSO and DE to search for the optimal mutation, which we refer to as PSO-DE.
- The algorithm is designed to preserve the strengths of both the algorithms.
- The vanilla PSO is particularly prone to get stuck in local optima, Hence, DE is introduced to increase the exploration ability of the search algorithm.

IoT Testbed


Wireless Connection
Wired Connection
Switch
Internet
Monitor Laptop
Router
Attack Laptop
TP-Link Tapo C200
EZVIZ Mini O Plus
SKT NU 100
Google Home Mini
Phillips Hue Bridge

Datasets

- Two datasets: benchmark Kitsune dataset and dataset captured from our own IoT testbed
- The benign data
  - consists of packets generated via commands such as turning off light bulbs, playing music on Google Home, etc, over 30 minutes.
- The malicious data
  - consists of two main types of malicious attacks, probing and Denial of Service (DoS), all targeted at Google Home Mini.
  - Distributed attacks such as Mirai Botnet were excluded since it requires compromising other IoT devices, which is outside the scope of our threat model.
  - Probing attacks include Port Scan (PS) and OS Detection (OD), and we repeat PS twice and OD four times. DoS attacks include HTTP Flooding (HF) with LOIC at the highest intensity for 6 minutes.

Target DL/ML Algorithms and surrogate model

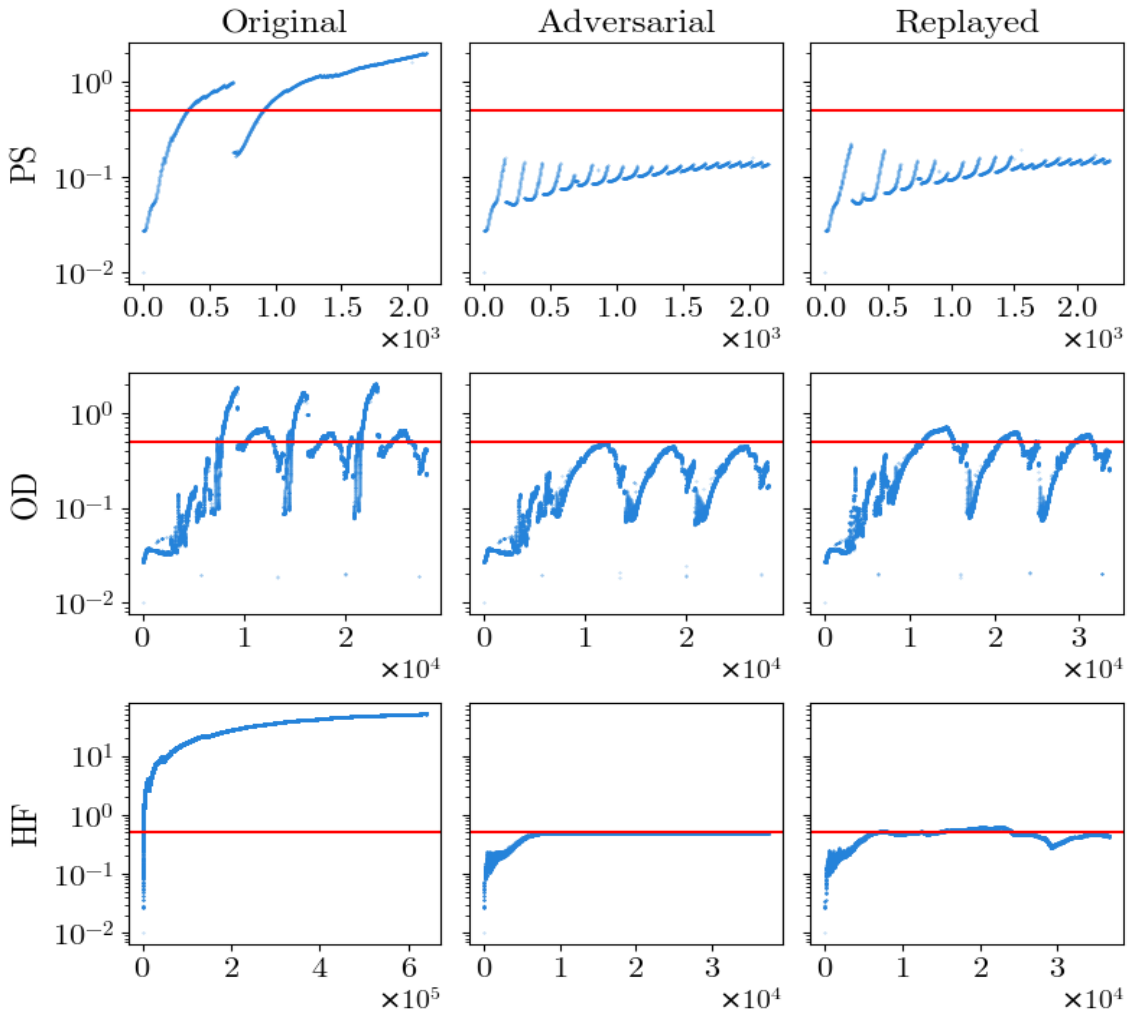
- Target/victim models
  - Kitsune, a state-of-the-art NIDS for IoT network
  - Self-Organising Maps (SOM) implemented with the python package MiniSOM.
  - Robust Random Cut Forest (RRCF) implemented with the python package rrcf.
  - Local Outlier Factor (LOF) implemented with sklearn.
  - One-Class SVM (OCSVM) implemented with sklearn.
  - Fast Random projection-based One-Class Classification (FROCC), the state-of-the-art One-Class classifier
- Surrogate model
  - a vanilla autoencoder written in TensorFlow 2
  - The encoder consists of three dense layers with 32, 8, and 2 neurons, respectively, and the Decoder consists of three dense layers with 8, 32, and 100 neurons. All layers use the ReLU activation function except for the last layer of Decoder, where it uses Sigmoid.
  - The architecture of the surrogate model is arbitrarily chosen, and we intentionally did not conduct any hyperparameter search to find the optimal structure.
  - The surrogate model is trained on all benign packets with one epoch to mimic online detection.
  - The threshold of the surrogate model is determined as three standard deviations away from the mean of the anomaly scores on benign data.

Metrics

- Performance Metrics
  - True Negative Rate (TNR): measures the ratio of packets in benign traffic correctly identified as benign packets
  - Malicious Detection Rate (MDR): measures the ratio of packets classified as malicious in the malicious traffic.
    - MDR is similar to True Positive Rate (TPR), but since we do not have ground truth labels, we use MDR instead
- Evasion Metrics
  - Detection Rate (DR) and Evasion Rate (ER), with each metric measured for adversarial traffic (ADR and AER) and replayed traffic (RDR and RER).
  - DR measures the ratio of the adversarial/replayed packets that have been classified as malicious and gives an indication of the robustness of NIDS under adversarial/replayed traffic (higher is better for defender).
  - ER measures the percentage of the adversarial/replayed packets that evade detection compared to the original attack, indicating the effectiveness of the adversarial attack (higher is better for attacker).
- Semantic Metrics
  - Semantic metrics compare the severity of the adversarial traffic on the target system to the original, unmodified attack
  - For DoS attacks, we measure the Round-Trip Time (RTT) and calculate the Relative Round-trip Delay (RRD) of the device under flooding attacks compared to the normal environments.
  - For Probing attacks, we compare Ports Scanned (PS), the Relative Ports Scanned (RPS) and Relative Time Delay (RTD) between the adversarial and unmodified attacks.

Experimental Results

- Anomaly score of three types of traffic
  - Original: original detectable attack
  - Adversarial: theoretically generated adversarial packets
  - Replayed: replay of adversaria packet
- Adversarial packets are evasive
- Replayed is not
  - Due to inherent propagation delays, the arrival time will not be exact



Attacks
- Port Scan (PS)
- OS Detection (OD)
- HTTP Flooding (HF)

Experimental Results - Evasiveness of Adversarial Traffic

- Replayed traffic is less evasive compared to adversarial traffic, indicated by RER being higher than AER for all attacks.
  - Adversarial traffic Evasion Rate (AER)
  - Replayed traffic Evasion Rate (RER)
- Kitsune and FROCC are highly vulnerable to LM, but other anomaly detectors are not.
  - Adversarial traffic Detection Rate (ADR) of LM with Kitsune and FROCC is 0 for all three attacks, while other detection algorithms have ADR higher than 0.

Table 1: DR and ER of the adversarial/replayed traffic generated with Liuer Mihou against various NIDS.

Attack	ML	TNR	MDR	ADR	RDR	AER	RER
PS	Kitsune	1.000	0.74	0.00	0.00	1.00	1.00
PS	SOM	0.993	0.89	0.05	0.15	0.94	0.83
PS	LOF	0.999	0.98	0.98	0.98	0.00	0.00
PS	RRCF	0.992	0.93	0.71	0.71	0.24	0.23
PS	OCSVM	0.999	0.99	0.98	0.99	0.01	0.00
PS	FROCC	1.000	0.50	0.00	0.00	1.00	1.00
OD	Kitsune	1.000	0.41	0.00	0.25	1.00	0.39
OD	SOM	0.993	0.66	0.29	0.53	0.55	0.19
OD	LOF	0.999	0.95	0.95	0.97	0.00	-0.01
OD	RRCF	0.995	0.64	0.46	0.55	0.28	0.15
OD	OCSVM	0.999	0.84	0.83	0.88	0.00	-0.05
OD	FROCC	1.000	0.19	0.00	0.07	1.00	0.63
HF	Kitsune	1.000	1.00	0.00	0.33	1.00	0.67
HF	SOM	0.993	1.00	0.91	0.77	0.09	0.23
HF	LOF	0.999	1.00	1.00	1.00	0.00	0.00
HF	RRCF	0.978	1.00	1.00	1.00	0.00	0.00
HF	OCSVM	0.999	1.00	1.00	1.00	0.00	0.00
HF	FROCC	1.000	1.00	0.00	0.01	1.00	0.99

Experimental Results - Maliciousness of Adversary

- HTTP Flooding (HF) attacks (Table 4)
  - The goal is to make the Google Home device unresponsive, so we measure the average round trip time using ping
  - The delay in RTT caused by the replayed HTTP flooding attack is only a small proportion of the unmodified HTTP flooding, but it is still larger than normal
  - the increase of RTT of the adversarial traffic generated with LM was barely noticeable.
- OS Detection (OD) attacks (Table 5)
  - can fully scan all the ports scanned by the unmodified attack with less than a ten percent increase in Relative Time Delay (RTD) for LM.
- Port Scan (PS) attacks (Table 5)
  - can scan over 90% of the original ports but takes ten times more time than the original Port Scan attack.

Table 4: Comparison between original and adversarial HTTP Flooding attacks.

Traffic Type	RTT (ms)	RRD
Normal	6.602	1.000
Original HF	173.393	26.264
Liuer Mihou HF	10.303	1.561

Table 5: Comparison between original and adversarial scanning attacks.

Traffic	Ports Scanned	RPS	Time(s)	RTD
Original OD	155	1	1,654.55	1.00
Liuer Mihou OD	155	1	1,792.52	1.08
Original PS	155	1	1.99	1.00
Liuer Mihou PS	150	0.968	19.99	10.04

Against Adversarial Defence

- Introduce two adversarial detectors as defence
  - Feature Squeezing (FS)
  - Mag-Net
- The detectors first detect if the input is adversarial or clean, then the NIDS detects the input as benign or malicious.
- FS cannot detect the adversarial examples
- Mag-Net detects the adversarial examples, but the reforming stage makes the malicious examples evade detection

Table 9: Detection result for Kitsune with Feature Squeezing.

	FS Prediction		Kitsune Prediction		
Traffic	**Clean**	**Adv.**	**Benign**	**Malicious**	**Total**
Benign	14400	0	14400	0	14400
PS	1975	167	565	1577	2142
PS_{adv}	2142	0	2142	0	2142
PS_{rep}	2252	0	2252	0	2252
OD	27075	949	16651	11373	28024
OD_{adv}	28074	0	28074	0	28074
OD_{rep}	33571	0	33571	8313	33571
HF	33462	607305	935	639832	640767
HF_{adv}	37440	0	37440	0	37440
HF_{rep}	36653	0	36653	12089	36653

Table 10: Detection result for Kitsune with Mag-Net.

	Mag-Net Prediction		Kitsune Prediction		
Traffic	**Clean**	**Adv.**	**Benign**	**Malicious**	**Total**
Benign	14400	0	14400	0	14400
PS	126	2016	2142	0	2142
PS_{adv}	750	1392	2142	0	2142
PS_{rep}	702	1550	2252	0	2252
OD	7400	20624	27789	235	28024
OD_{adv}	9662	18412	28074	0	28074
OD_{rep}	7674	25897	33571	0	33571
HF	322	640445	640118	649	640767
HF_{adv}	690	36750	37440	0	37440
HF_{rep}	728	35925	36653	0	36653

Some challenges

- Data issues
  - No standard AE benchmark dataset for NIDS
  - Structured Data - cannot freely modify the features
  - Sequentially Related - the same packet will produce different features depending on previous packets
  - published dataset is i) not general and has to be anonymised, ii) often have truncated payload
- Transferability: Evasiveness and Maliciousness trade-off
  - Liuer Mihou is a transfer-based attack that leverages adversarial transferability across ML/DL algorithms.
  - Since knowing the actual detection model is infeasible in practice, the similarity of the decision boundary cannot be increased. Therefore, to increase evasiveness, the attacker have to rely on lowering the threshold value of the surrogate model.
  - lowering the surrogate threshold will inevitably lower the maliciousness of the adversarial traffic.
- Weakness of Adversarial Defence
  - We used two plug-and-play adversarial defence methods: Feature Squeezing and Mag-Net. These defences were designed initially for classification algorithms, and the results from our experiment have shown that both methods are unsuitable in the NIDS domain.
  - First, packet-level attacks make large changes in the input feature and have realistic distribution as benign traffic, and adversarial detectors that intentionally ignore small perturbations, such as Feature Squeezing, fail to detect the adversarial examples.
  - Second, only benign traffic is available to train the adversarial detectors under a realistic NIDS threat model. Therefore, the adversarial detector cannot distinguish between adversarial and malicious traffic, classifying all attacks as adversarial.

Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

On-going research

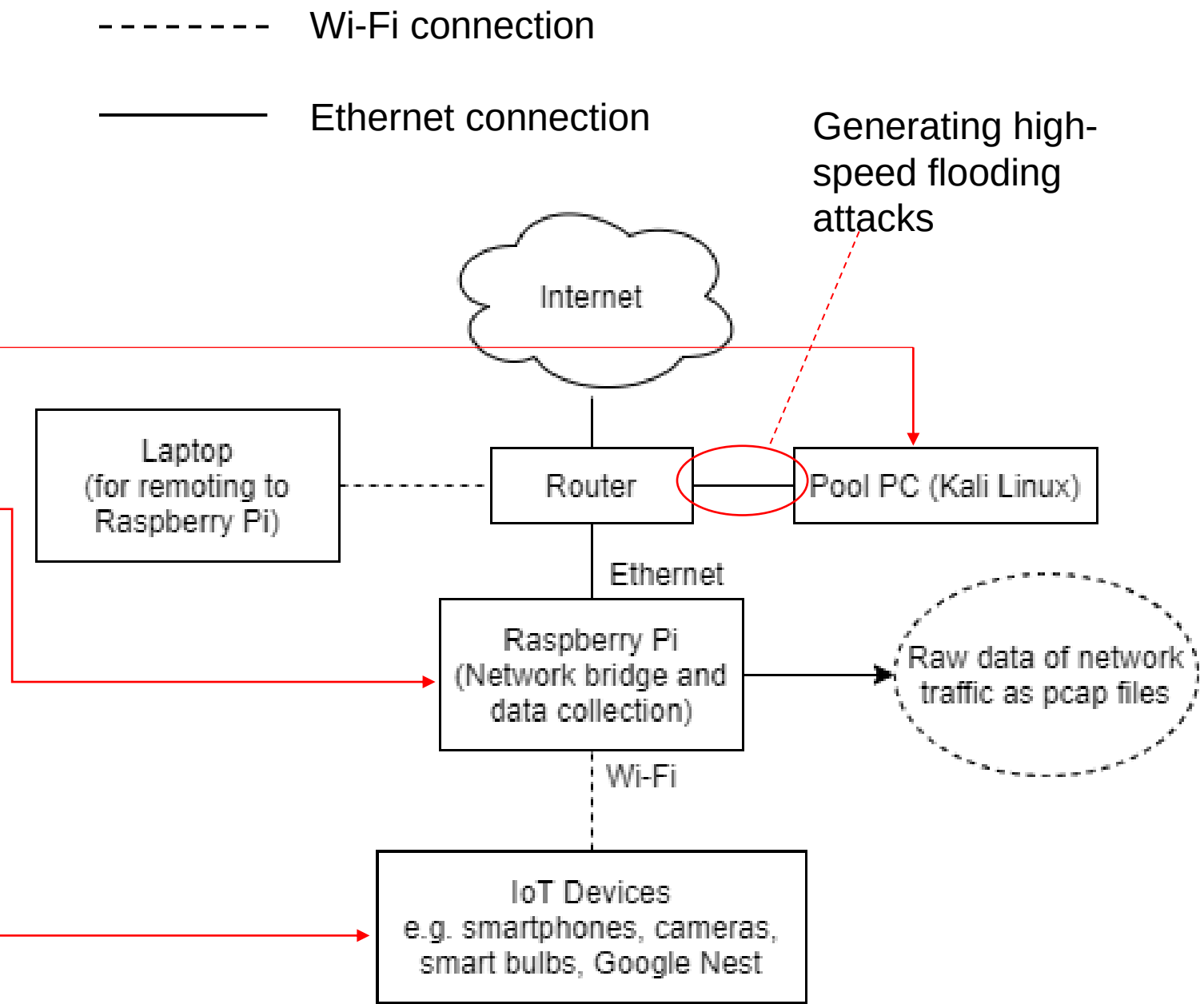
- UQ IoT testbed and data collection
  - Initial data is available
  - Additional datasets will be collected

- Attacks
  - Practical and Replayable Evasion Attacks for Deployed ML/DL based IDS in UQ IoT network.
  - AML based evasion attacks for Autonomous vehicle networks (CANBus)

- Defences
  - MTD based adversarial defences
  - Explainability (e.g., Feature to class /Decision boundary visualization)

UQ IoT Testbed Setup

- Pool PC (Kali Linux) – Attacker:
  - Generate cyber-attacks
- Raspberry Pi:
  - Network Bridge and Wi-Fi AP
  - Data Collection
- IoT devices:
  - Generate benign samples
  - Attack targets



List of IoT devices

No.	Device Name	No.	Device Name
1	Smartphone 1	2	Smartphone 2
3	Smart Bulb 1	4	Smart Bulb 2
5	Smart Clock 1	6	Smart Clock 2
7	IP Camera 1	8	IP Camera 2
9	Google Nest Mini 1	10	Google Nest Mini 2
11	Smart Plug 1	12	Smart Plug 2
13	Smart TV	14	Telnet Raspberry Pi
15	Bridge Raspberry Pi	16	Pool PC
17	Router	18	Laptop

Data collection

Generate Network Packets from:

- IoT devices – Benign Samples
- Attacker – Attack Samples

Benign Samples

- Streaming videos and music
- Checking social media
- Adjusting brightness and colours of bulbs
- *etc.*

❑7 days of data mimicking daily activities of devices for a whole week
  - 5 weekdays
  - 1 Saturday
  - 1 Sunday

Data collection

Generate Network Packets from:

- IoT devices – Benign Samples
- Attacker – Attack Samples

Attack Samples

- 9 types of cyber-attacks:

Host Discovery	Port Scanning	Service Detection
ARP Spoofing	Telnet Brute-force	SYN Flooding
ACK Flooding	HTTP Flooding	UDP Flooding

❑ Collected separately for each device under each attack type (convenient for labelling)

Dataset Summary

- **Features**: 107
  No., Date_Time, Src_Port, Dest_Port, Protocol, Length, Label, 100 extracted features
- **Size**: 70.8GB

Label	Number of Packets	Percentage
Normal	22480614	54.294%
UDP Flooding	8392711	20.270%
ACK Flooding	6141155	14.832%
SYN Flooding	3923876	9.477%
HTTP Flooding	402252	0.972%
Service Detection	38908	0.094%
Port Scanning	16519	0.040%
Host Discovery	6120	0.015%
Telnet Brute-force	2167	0.005%
ARP Spoofing	661	0.002%
Total	41404983	-

He, Ke, Kim, Dan, Zhang, Zhien, Ge, Mengmeng, Lam, Ulysses, and Yu, Jiaqi(2022). *UQ IoT IDS dataset 2021*. The University of Queensland. Data Collection.https://doi.org/10.48610/17b44bb

Building ML/DL models for NIDS

- Comparison with Kitsune

	CNN	RNN	RF	Kitsune
Accuracy	99.62%	99.65%	99.98%	94.99%
Training Time	50 min	700 min	1 min	2 min
False Positive Rate	0.28%	0.19%	0.02%	0.47%

- Supervised Learning Models: lower False Positive Rates, higher Accuracy
- Kitsune: much faster than Neural Networks

On-going research

- UQ IoT testbed and data collection
  - Initial data is available
  - Additional datasets will be collected

- Attacks
  - Practical and Replayable Evasion Attacks for Deployed ML/DL based IDS in UQ IoT network.
  - AML based evasion attacks for Autonomous vehicle networks (CANBus)

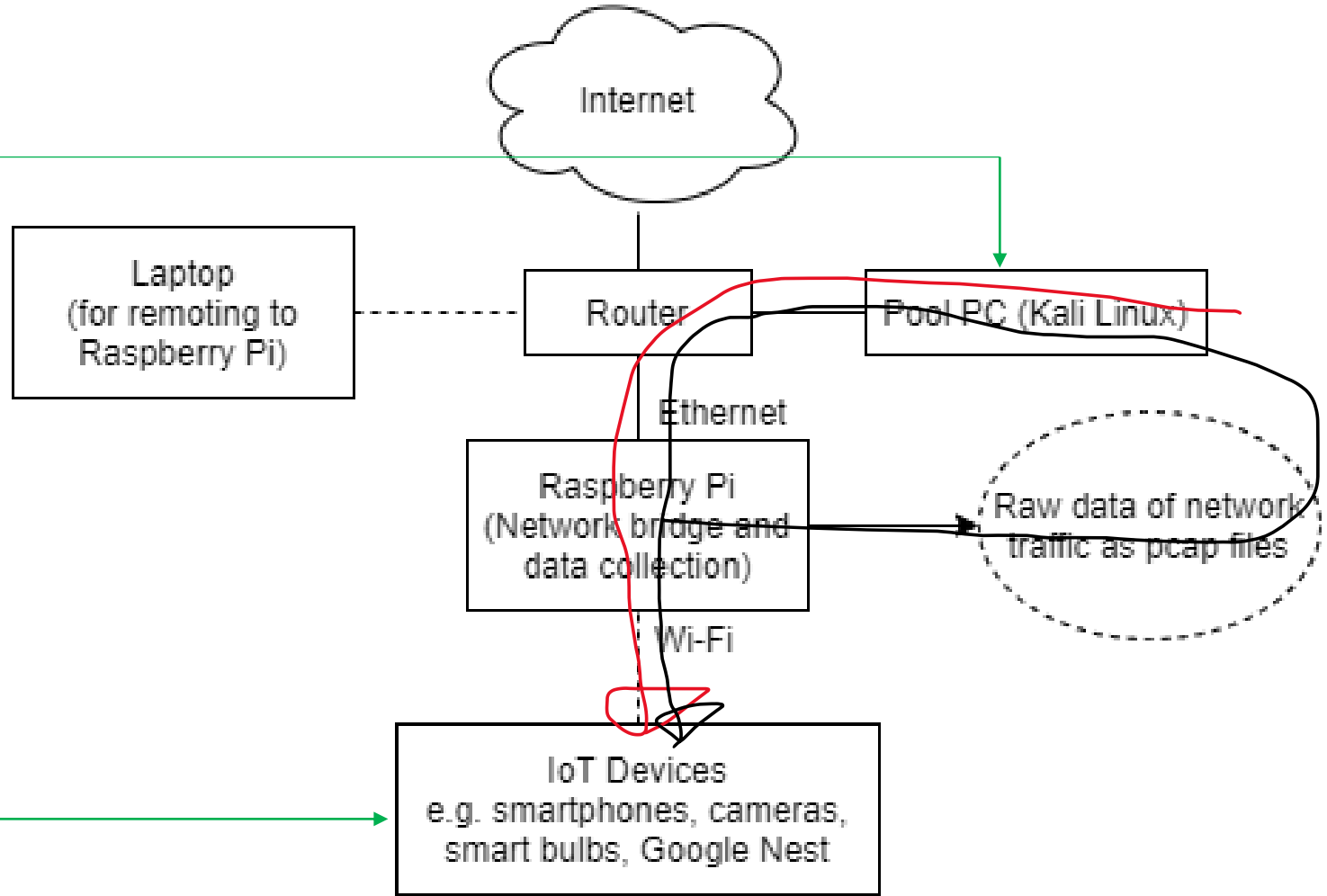
- Defences
  - MTD based adversarial defences
  - Explainability (e.g., Feature to class /Decision boundary visualization)

On-going work

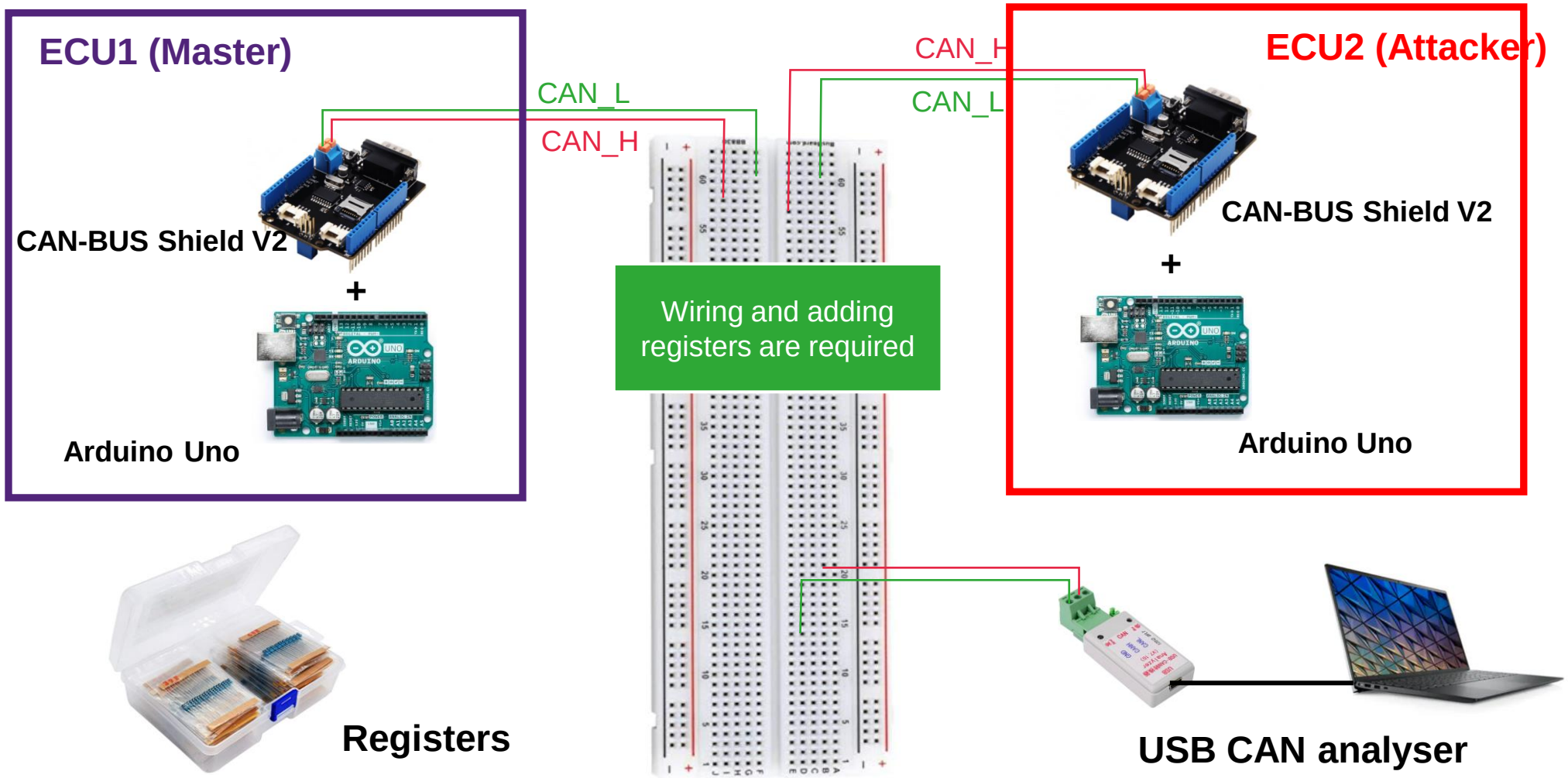
- LM+: Practical, Replayable and Real-time Evasion Attacks for Deployed ML/DL based IDS in UQ IoT network.
  - Evasiveness
  - Maliciousness (Attack impact)

- Pool PC (Kali Linux) – Attacker:
  - Generate cyber-attacks

- IoT devices:
  - Generate benign samples
  - Attack targets



UQ In-Vehicle Testbed



On-going research

- UQ IoT testbed and data collection
  - Initial data is available
  - Additional datasets will be collected

- Attacks
  - Practical and Replayable Evasion Attacks for Deployed ML/DL based IDS in UQ IoT network.
  - AML based evasion attacks for Autonomous vehicle networks (CANBus)

- Defences
  - MTD based adversarial defences
  - Explainability (e.g., Feature to class /Decision boundary visualization)

MTD based adversarial defenses

- What to move
  - We are operating in a rather limited space
  - Consider changing the detection algorithm and/or feature extractor
- How to move
  - Redundancy is not really applicable in the context of adversarial defence
  - Focus on shuffling and diversity
- When to move
  - Either change the model on a per query basis or per time interval
  - Moving a target does not incur much cost, thus we can move frequently

- Previous studies in MTD inspired adversarial defence have broken down “how to move” into three stages
  - Model diversification - generate diverse range of models
  - Model selection - select all or partial models to be used in classification
  - Model detection - how to use the selected models to classify the input

Process of MTD – on-going work

- Trains a pool of models with
  - Feature mapper mutation - randomly alter kitsunes feature mapper
  - Model mutation - use different anomaly detectors
  - Parameters mutation - perturb weights and biases of a trained kitsune model
  - Training data mutation - use a subset of data to train each model, can be done in conjunction with other mutation techniques
- Evaluate each model on benign data to calculate threshold, and remove models that have lower TNR than the original model
  - Note this will not filter out models that classify everything as benign (e.g. IF and some OCSVM models)
- During evaluation, a random model is picked to process every 1000 packets

Explainability vs. interpretability

- Lipton (2018) and Rudin (2019) provides similar definitions:

- Interpretability provides transparency and answers “How does the model work?”
  - Intrinsically explainable models can be considered transparent. However, it requires the features to be meaningful

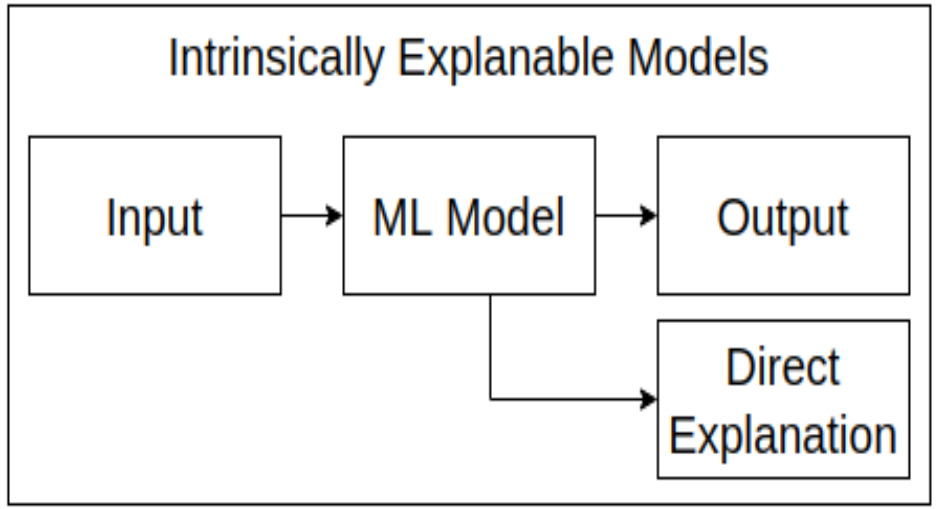
- Explainability answers “What else can the model tell me?”
  - Post hoc explanations with textual or visual explanations

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 16(3), 31-57.
Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

Existing Explainability Methods

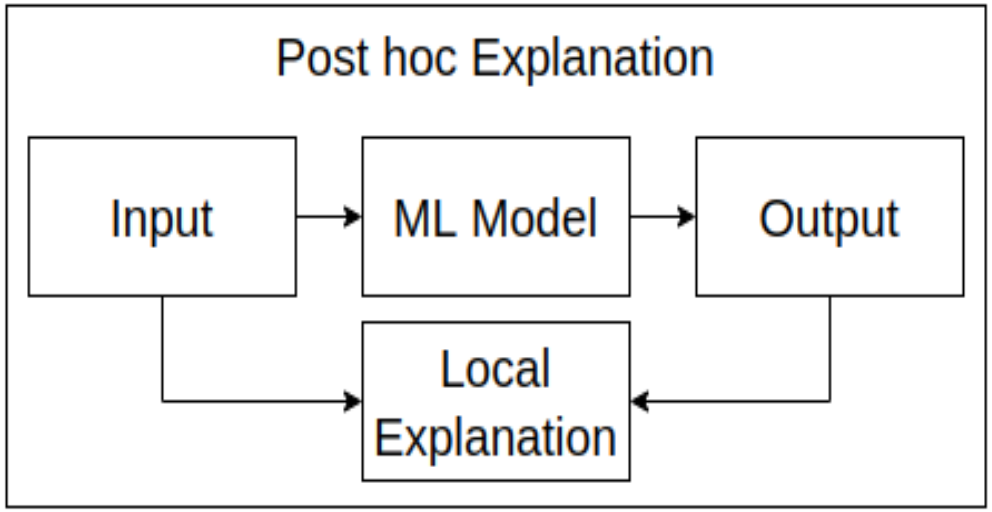
Intrinsically explainable models

- Simple models where the decision function is directly explainable
  - e.g. decision tree, linear regression
- The features have to be informative



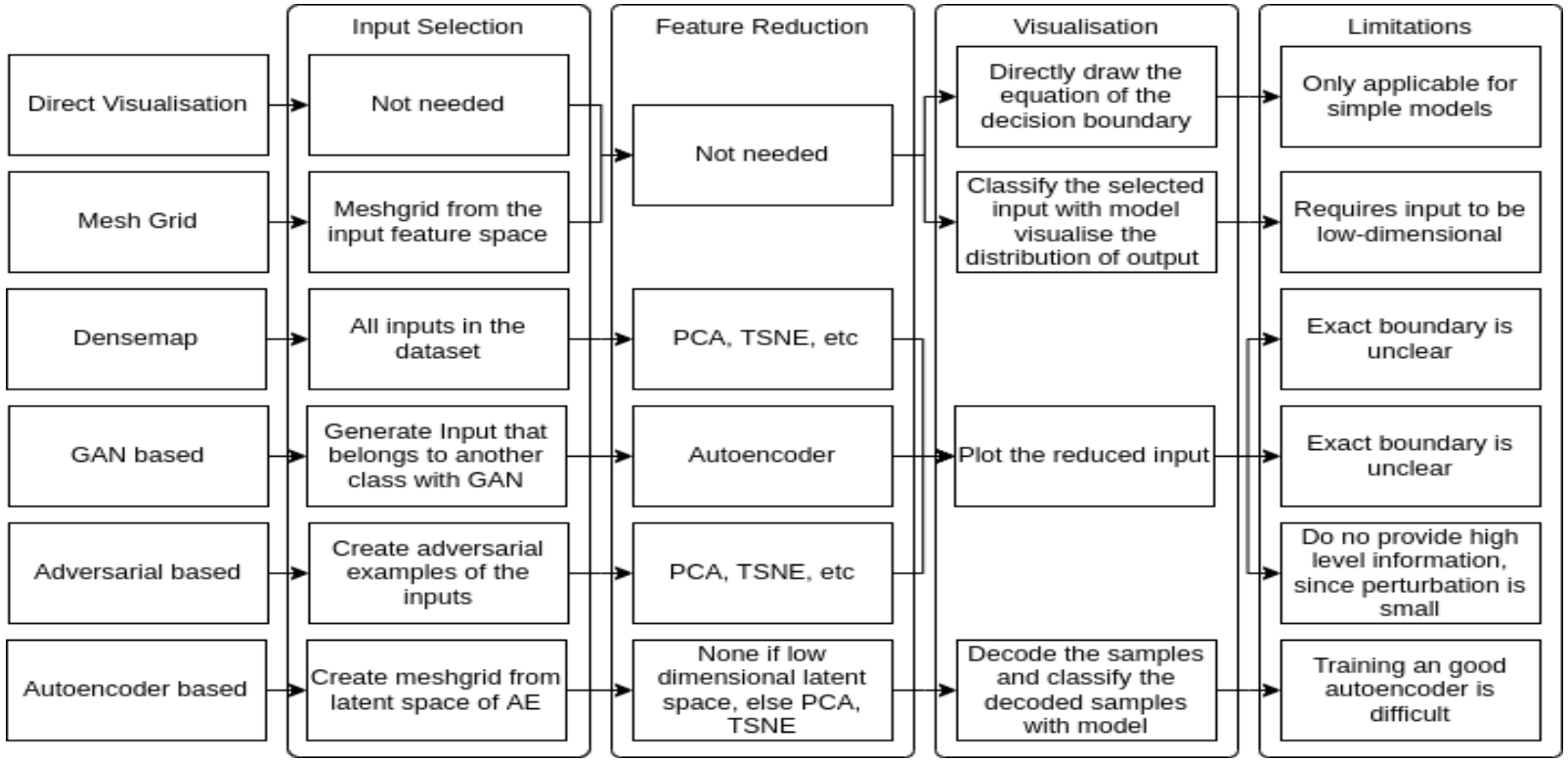
Post hoc explanation methods

- Complex models where the decision function is high dimensional
- Simplify local area of the model with linear function that are explainable.
  - e.g., SHAP, LIME, DeepLIFT
- Visualise features with dimensionality reduction



Explainability – some questions to answer

- Which features contribute most towards classifying a malicious packet as abnormal?
- Which features contributes most towards the sequence of packets?
- How can we visualise the decision boundary of the NIDS?



Outline

- About UQ/me
- IDS overview
- IDS^2 project overview
- AML for NIDS: a survey
- Practical Evasion Attacks for NIDS
- On-going work
- Q&A

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Thank you! Q&A

Adversarial Machine Learning in Network Intrusion Detection System

Associate Professor Dan Kim

Deputy Director of UQ Cyber

dan.kim@uq.edu.au

Take aways

- AI (ML/DL) can be used to detect network intrusion.
- Evasion attacks can be generated at network packet level.
- More practical attacks need to be developed.
- Defence against Adversarial attacks has to be developed more.

My publications in AI security/security for AI (1/3)

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

- [WISC02/KISC02] J. Lee, D. S. Kim, J. S. Park, D. B. Yeom, Detecting Host-based Intrusion with SVM classification, KISC2002 (in Korean)
- [ICOIN03] Dong Seong Kim, Jong Sou Park: Network-Based Intrusion Detection with Support Vector Machines. ICOIN 2003: 747-756
- [W2GIS04] Jong Sou Park, Hong Tae Jin, Dong Seong Kim: Intrusion Detection System for Securing Geographical Information System Web Servers. W2GIS 2004: 110-119
- [AINA05] Dong Seong Kim, Ha-Nam Nguyen, Jong Sou Park: Genetic Algorithm to Improve SVM Based Network Intrusion Detection System. AINA 2005: 155-158
- [CISC05] Jong Sou Park, Khaja Mohammad Shazzad, Dong Seong Kim: Toward Modeling Lightweight Intrusion Detection System Through Correlation-Based Hybrid Feature Selection. CISC 2005: 279-289
- [MICAI06] Dong Seong Kim, Sang Min Lee, Jong Sou Park: Toward Lightweight Detection and Visualization for Denial of Service Attacks. MICAI 2006: 632-640
- [CIS07] Lee, Sang Min, Kim, Dong Seong, and Park, Jong Sou (2007). A hybrid approach for real-time network intrusion detection systems. 2007 International Conference on Computational Intelligence and Security, CIS 2007, Harbin, China, 15-19 December 2007. Piscataway, NJ, United States: IEEE.https://doi.org/10.1109/CIS.2007.28
- [ARES08] Dong Seong Kim, Muhammad Anwarul Azim, Jong Sou Park:Privacy Preserving Support Vector Machines in Wireless Sensor Networks. ARES 2008: 1260-1265
- [ISA09] Tae Hwan Kim, Dong Seong Kim, Sang Min Lee, Jong Sou Park: Detecting DDoS Attacks Using Dispersible Traffic Matrix and Weighted Moving Average. ISA 2009: 290-300
- [PRDC09] Sang Min Lee, Dong Seong Kim, YoungHyun Yoon, Jong Sou Park: Quantitative Intrusion Intensity Assessment Using Important Feature Selection and Proximity Metrics. PRDC 2009: 127-134
- [CISIS10] Sang Min Lee, Dong Seong Kim, Ji Ho Kim, Jong Sou Park: Spam Detection Using Feature Selection and Parameters Optimization. CISIS 2010: 883-888

My publications in AI security/security for AI (2/3)

- [IMIS11] Je Hak Lee, Dong Seong Kim, Sang Min Lee, Jong Sou Park: DDoS Attacks Detection Using GA Based Optimized Traffic Matrix. IMIS 2011: 216-220
- [JISIS12] Sang Min Lee, Dong Seong Kim, Jong Sou Park: A Survey and Taxonomy of Lightweight Intrusion Detection Systems. J. Internet Serv. Inf. Secur. 2(1/2): 119-131 (2012)
- [ICA3PPW15] Hong, Jin B., et al. "Scalable network intrusion detection and countermeasure selection in virtual network systems." International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2015.
- [ICSSA18] Taehoon Eom, Heesu Kim, SeongMo An, Jong Sou Park and Dong Seong Kim, Android Malware Detection using Feature Selections and Random Forest, International Conference on Software Security and Assurance (ICSSA 2018), July 26-27, Seoul, South Korea
- [Trustcom19m] Ke He, Dong Seong Kim: Malware Detection with Malware Images using Deep Learning Techniques. TrustCom/BigDataSE 2019: 95-102
- [IEEEA20] Jaehyoung Park, Dong Seong Kim, Hyuk Lim: Privacy-Preserving Reinforcement Learning Using Homomorphic Encryption in Cloud Computing Infrastructures. IEEE Access 8: 203564-203579 (2020)
- [SPIE20] Yoon, S., Cho, J. H., Kim, D. S., Moore, T. J., Nelson, F. F., Lim, H., ... & Kamhoua, C. (2020, April). Moving target defense for in-vehicle software-defined networking: IP shuffling in network slicing with multiagent deep reinforcement learning. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II (Vol. 11413, pp. 617-626). SPIE.
- [IEEEA21a] Hyunjun Kim, Sunwoo Ahn, Whoi Ree Ha, Hyunjae Kang, Dong Seong Kim, Huy Kang Kim, Yunheung Paek: Panop: Mimicry-Resistant ANN-Based Distributed NIDS for IoT Networks. IEEE Access 9: 111853-111864 (2021)
- [WISA21] Raymond Mogg, Simon Yusuf Enoch, Dong Seong Kim: A Framework for Generating Evasion Attacks for Machine Learning Based Network Intrusion Detection Systems. WISA 2021: 51-63
- [IEEEA21b] Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J. Moore, Frederica Free-Nelson, Hyuk Lim: DESOLATER: Deep Reinforcement Learning-Based Resource Allocation and Moving Target Defense Deployment Framework. IEEE Access 9: 70700-70714 (2021)

My publications in AI security/security for AI (3/3)

- [Mogg20] Generating Evasion Attacks Against Intrusion Detection Systems Using Genetic Algorithms, UQ ITEE thesis (honours) 2020, S1
- [Baxter20] Decision tree based network intrusion detection systems, UQ ITEE thesis (honours) 2020, S2
- [Lee20] Disguising an Attack Against a Clustering based Network Intrusion Detection System, UQ ITEE thesis 2020 (honours), S2
- [Amalia20] Performance Comparison of Adversarial Attacks on Intrusion Detection System, UQ ITEE thesis (honours) 2020, S2
- [Liu21] Genetic Algorithm based Adversarial Examples Generation against Machine Learning based Intrusion Detection Systems, UQ ITEE thesis 2021 (master), S1
- [Wang21a] Semantic Preserving Adversarial Attack Generation with Autoencoder and Genetic Algorithm, UQ ITEE thesis 2021 (master), S1
- [Stanwix21] Cyber Defence Generation and Automation, UQ ITEE thesis 2021 (honours), S1
- [Zhou21] Markov Decision Process for Automatic Cyber Defense, UQ ITEE thesis 2021 (honours), S1
- [Wang21b] Intrusion Detection and Response using AI Planning, UQ ITEE thesis (honours) 2021, S2
- [Chancellor21] Cyber Attacks against Deployed Machine Learning based Intrusion Detection Systems, UQ ITEE thesis (honours) 2021, S2
- [Xia21] Privacy Preserving Machine Learning Implementation and Testing, 2021, UQ ITEE thesis (master) 2021, S2
- [Hu22] Practical Adversarial Attacks Generation for Black Box Machine Learning Models, UQ ITEE thesis (master), 2022, S1
- [Zhang22] Practical Cyber Attacks Generation against NIDS for IoT Netwoks, UQ ITEE thesis (honours) 2022, S1
- [Li22] Automated Pen Testing using Improved Reinforcement Learning Techniques, UQ ITEE thesis (honours) 2022, S1

MTD vs Ensemble

Characteristic	Ensemble	MTD
Philosophy	Combine prediction of multiple models (often weak) to become more robust	Constantly switch between (strong) models to make the model harder to attack
Goal	Create diverse set of models	Create diverse set of models
Output	Average / majority vote	The chosen model's output
Difficulty for attacker	Fool more than half of the ensemble	Fool all of the models