Alchemy: Stochastic Data Augmentation for Malicious Network Traffic Detection

NTT

Bo HU

Agenda

topic = background(**nw & security & ml**)

for each in topic.**use_cases**(3) :

[each.**data**, each.**features**]

try:

topic.**solve**(**problems**)

except:

pass

while Time:

topic.**future_works**

Background 1/3

- The botnet is not only a threat to IT services / organizations, but also became a serious problem to the whole Internet / society.


THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate WSJ. Magazine

Cyberattack Knocks Out Access to Websites

Popular sites such as Twitter, Netflix and PayPal were unreachable for part of the day

https://www.wsj.com/articles/denial-of-service-web-attack-affects-amazon-twitter-others-1477056080

A Report to the President

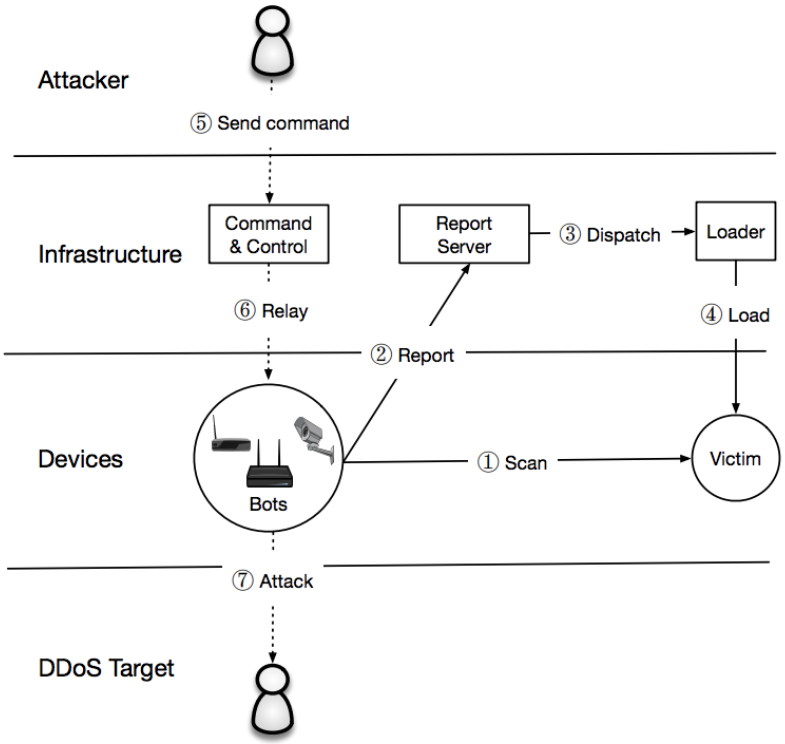
on

Enhancing the Resilience of the Internet and Communications Ecosystem Against Botnets and Other Automated, Distributed Threats

https://csrc.nist.gov/publications/detail/white-paper/2018/05/30/enhancing-resilience-against-botnets--report-to-the-president/final

Background 2/3

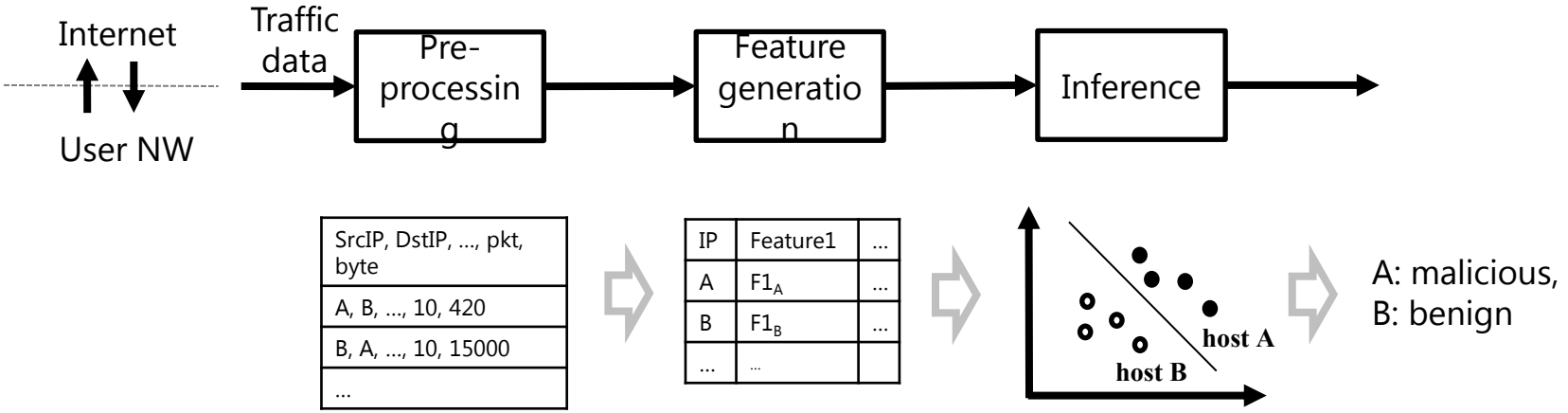
- Large-scale botnets consist of various components such as bots, loaders and Command and Control (C&C) servers.
- Attackers tend to use multiple (and hierarchical) hosts to control the botnet for higher availability.



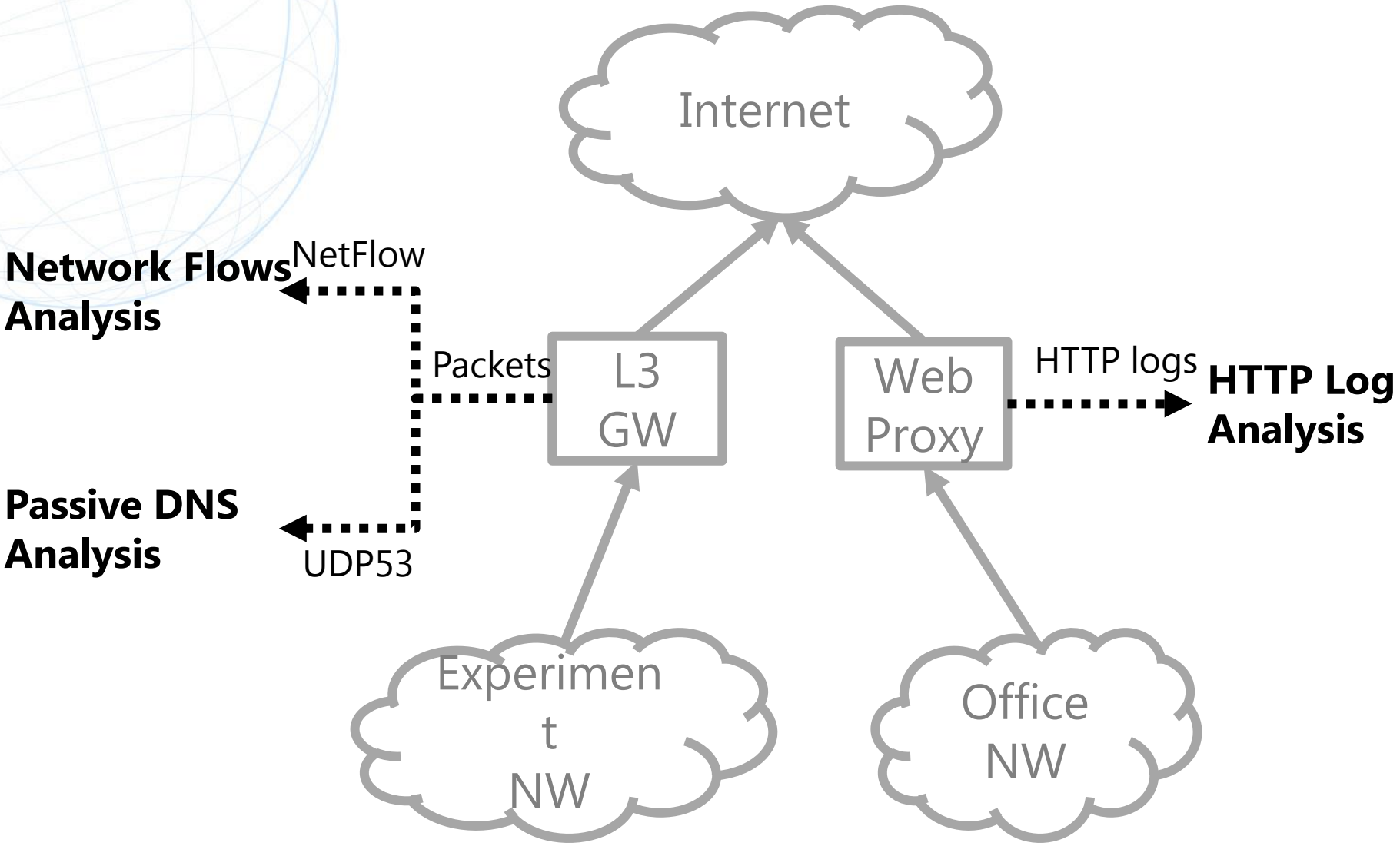
Framework/Operation of Mirai botnets [1]

Background 3/3

- Machine learning is a new hope for detecting botnet activities.
- Conventional methods are designed for detecting specific types of components (e.g., bot or C&C) [2][3]



Use Cases


Internet
NetFlow
Network Flows Analysis
Packets
L3 GW
Web Proxy
HTTP logs
HTTP Log Analysis
Passive DNS Analysis
UDP53
Experimen t NW
Office NW

Use Cases: Network Flows

NTT

Data

Timestamp	Protocol	Src IP	Dest IP	Src Port	Dest Port	packets	bytes	TCP flags
2017-08-*	TCP	*.*.7.58	*.*.4.90	59926	80	1	140	.AP...
2017-08-*	TCP	*.*.127.53	*.*.4.90	59513	80	1	40	.A....
2017-08-*	TCP	*.*.6.37	*.*.4.90	13729	80	1	46	.A....
2017-08-*	TCP	*.*.4.90	*.*.10.30	80	20113	1	40	.A.R..
2017-08-*	TCP	*.*.4.90	*.*.223.155	80	38615	1	1250	.A....
2017-08-*	TCP	*.*.4.90	*.*.245.31	80	58741	1	40	.A.R..

Feature Sets [2]

Feature Set	#	Feature Name
Flow Size	1	Packets per flow
	2	Bytes per packet
	3	Unique flow sizes
Client Access Patterns	4	Regular access patterns
	5	Unmatched flow density
Temporal	6	Flows per 5 minutes
	7	Clients per 5 minutes

Use Cases: HTTP Logs

Data

Dest IP	URL	Bytes out	Bytes in	method	User agent
..5.222	http://*.*.com/pr/72e8e276-8bc5-11e6-a5ec-0695da005429/assets/img/icon2-green.png	483	4215	GET	Mozilla/4.0
..5.222	http://*.*.com/pr/72e8e276-8bc5-11e6-a5ec-0695da005429/assets/img/icon1-green.png	483	3825	GET	Mozilla/4.0
..5.222	http://*.*.com/pr/public/js/detector.js	441	3128	GET	Mozilla/4.0
..27.174	http://www.*.com/collect	259	207	POST	-
..27.174	http://www.*.com/urchin.js	437	7210.0	GET	-
..27.174	http://www.*.com/collect?v=*&tid=*&aip=*&cid=*&t=*&ec=*&ea=*&el=*&sc=*	485	404.0	GET	-

Feature Sets [3]

Feature Set	#	Feature Name
FQDN, Path, Query	1	Length
	2	Digit ratio
	3	Lower/upper case ratio
	4	Ratio of digits
	5	Vowel changes ratio
	6	Ratio of a character with max occurrence
	7	Max length of consonant/vowel/digit stream
	8	Number of non-base64 characters
Other	9	Number of bytes in request
	10	Number of bytes in response
	11	Number of parameters in query
	12	Number of '/' in path/query/referrer

Use Cases: Passive DNS

Data

Timestamp	Query	Query type	Answers	TTLs
2016-03-*	js.*.*.com	A	js.*.*.com.*.*.com, *.*.210.110, *.*.210.120	2, 600, 600
2016-03-*	js.*.*.com	A	js.*.*.com.*.*.com	1800
2016-03-*	js.*.*.com	A	js.*.*.com.*.*.com, *.*.210.110, *.*.210.120	1800, 600, 600
2016-03-*	play.*.com	A	play.l.*.com, *.*.197.142	94, 120
2016-03-*	play.*.com	A	play.l.*.com, *.*.197.142	90, 116
2016-03-*	play.*.com	A	play.l.*.com, *.*.197.142	86, 112

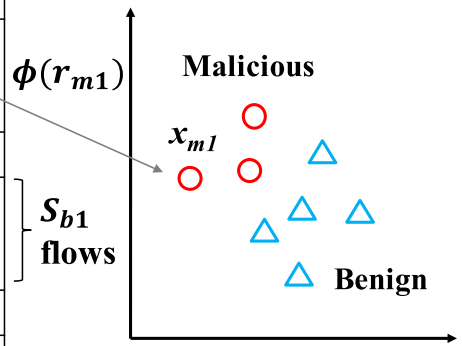
Feature Sets [4]

Feature Set	#	Feature Name
Temporal	1	Short life
	2	Daily similarity
	3	Repeating patterns
	4	Access ratio
DNS Answer	5	Number of distinct IP addresses
	6	Number of distinct countries
	7	Number of domains sharing the same IP address
TTL	8	TTL values
	9	Number of distinct TTL values
	10	Number of TTL change
	11	Percentage usage of specific TTL ranges
Domain	12	% of numerical characters

Problems

- Training an accurate model with
  - Few amount of malicious training sets
    - # of malicious < # of benign
  - Low quality statistical features values
    - Observed traffic of a malicious host is fewer than the benign. Fewer data, lower quality of statistics

Class	Target i	Bag r_i of network flows (ts, sip, dip, sp, dp, pr, pkt, byt)
Malicious	Server $m1$	8:00:00, $m1$, C_1, 80, 20000, 1, 100 8:00:00, C_2, $m1$, 30000, 80, 2, 500 …
	…	
Benign	Server $b1$	8:00:00, $b1$, C_1, 80, 20000, 1, 500 8:05:00, C_1, $b1$, 30000, 80, 1, 50 …
	…	
	…	



(1) Bagged raw data (2) Feature vectors

Contributions

- We propose a novel data argumentation method, Alchemy, to regenerate feature vectors for higher classification performance
- Alchemy not only increases training sets, but also enhance feature presentation, regardless of types of data, features and classifiers
- We evaluated effectiveness of Alchemy through three major types of real-world data

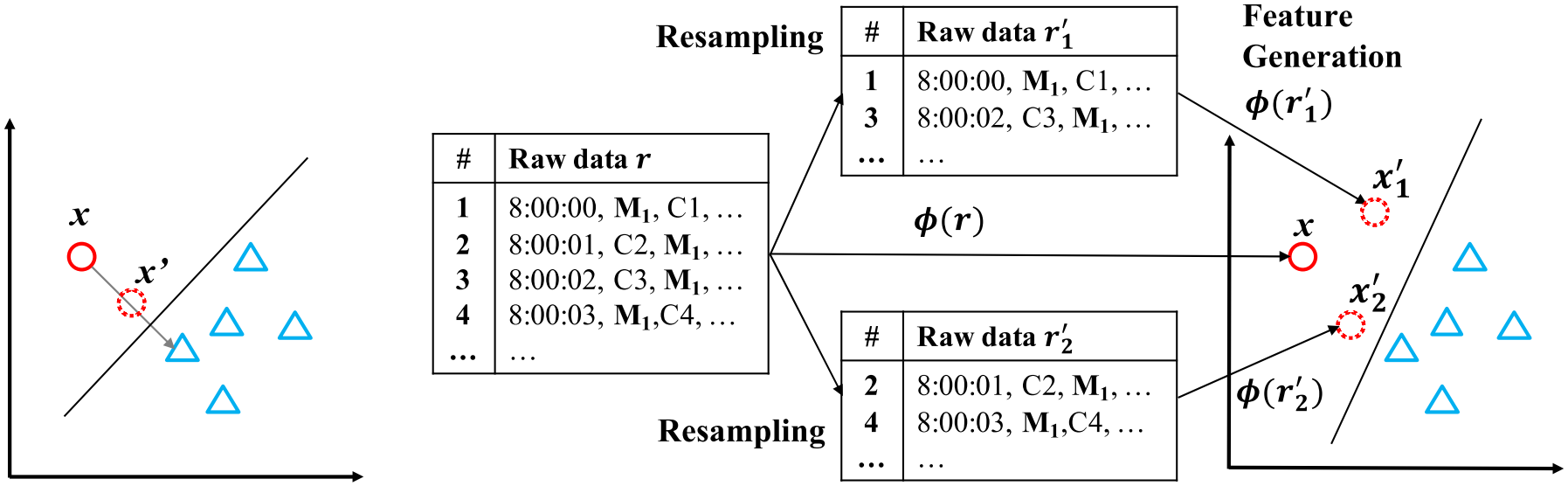
Related Works

- Synthetic data approach to increase training sets
  - Create new feature vectors based on existing ones or (random) distribution
  - Cannot represent nature of raw traffic
- Extra data approach to increase training sets
  - Utilize unlabeled data as labeled ones
  - Require more additional data from the real world

Approach	Technique	Summary	Level to apply
Synthetic data	Gaussian noise	Add Gaussian noise to each feature vector	Feature vector
	Domain knowledge-based	Extract several parts of a vector as new ones	
	SMOTE	Randomize new vectors between existing ones	
Extra real data	Semi-supervised learning	Treat unknown vectors as labeled	
	Active learning	Relect unknown vectors for labeling	
Reuse of data	**our proposal**	Resample raw data to create new feature vectors	Raw data

Proposal: Concept

- We propose a method (Alchemy) to resample several subsets of raw traffic for each host and regenerate multiple pseudo feature vectors to represent the original host in an ensemble way
  - Without any extra raw data
  - Regardless of data, features and classifiers

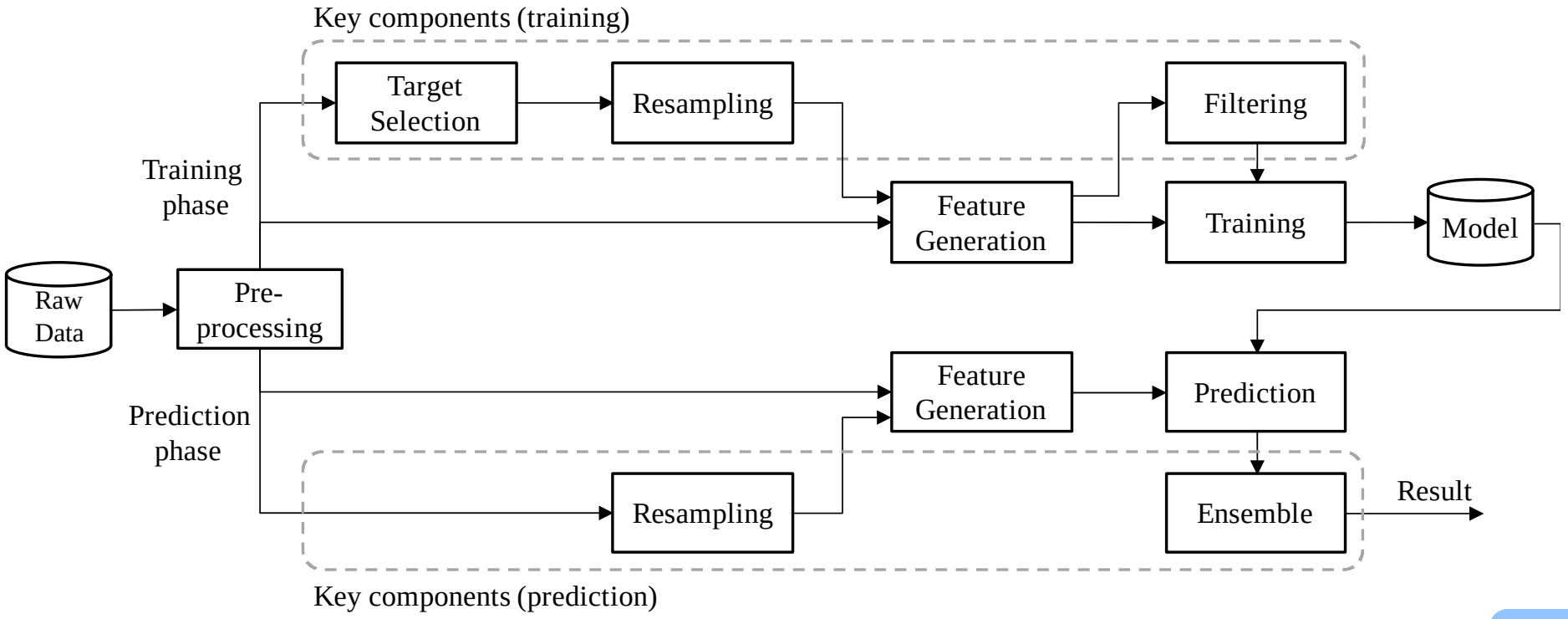


(1) Conventional (SMOTE)

(2) Alchemy

Proposal: System Design

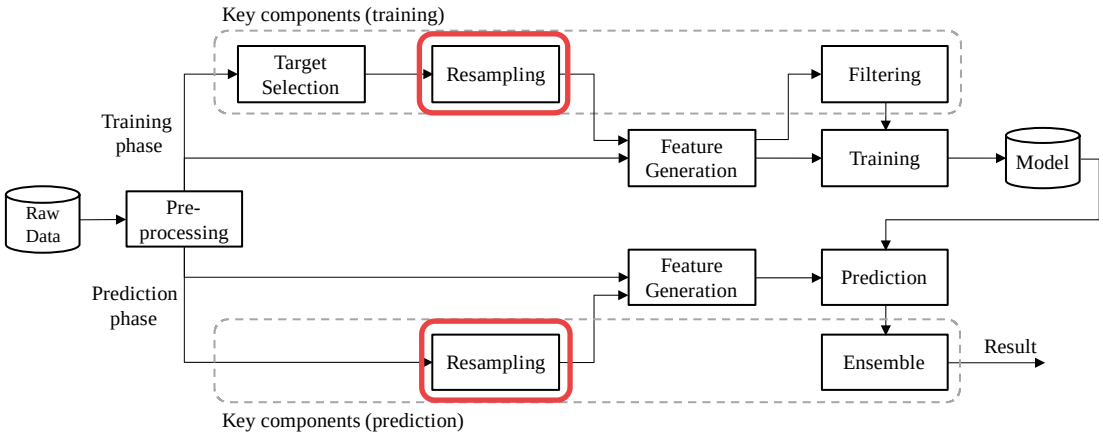
- Resampling can be applied in both of training and prediction
- Resampling rates can change with hosts or sampling operations
- Target selection and Filtering are components to control quality of pseudo feature vectors



Proposal: Resampling

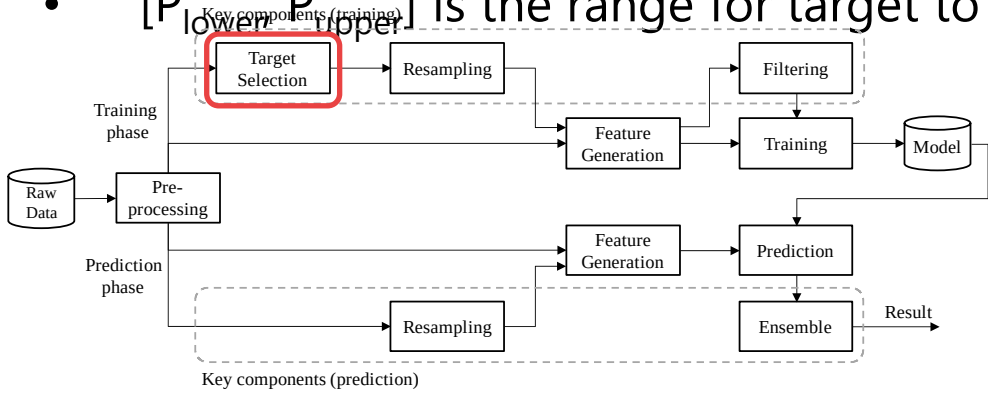
- Four options for sampling rates
- #3 and 4 use random rates

Approach	#	Option
Static	1	Static number
	2	One rate for all targets
Dynamic	3	One rate for one target
	4	Multiple rates for one target (one rate for one resampling operation)



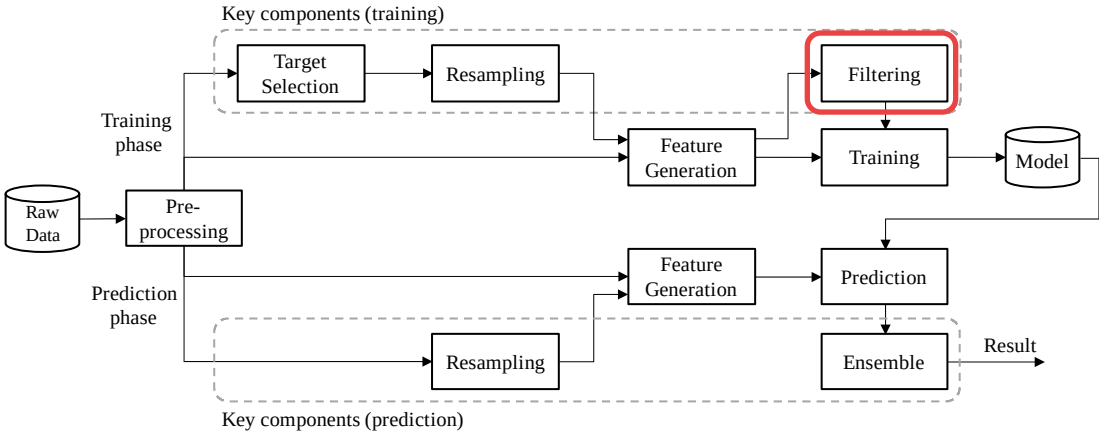
Proposal: Target selection

- Cross validation-based scoring strategy
  - Separate labeled data to training and validation sets
  - Training model with training sets
  - Predict targets in validation sets, and use the score of each target to decide whether to regenerate feature vectors
    - High score -> well feature presentation -> no needs to regenerate
    - Low score -> maybe outliers -> regenerated features may effect the model in a negative way
    - $[P_{lower}, P_{roper}]$ is the range for target to regenerate features



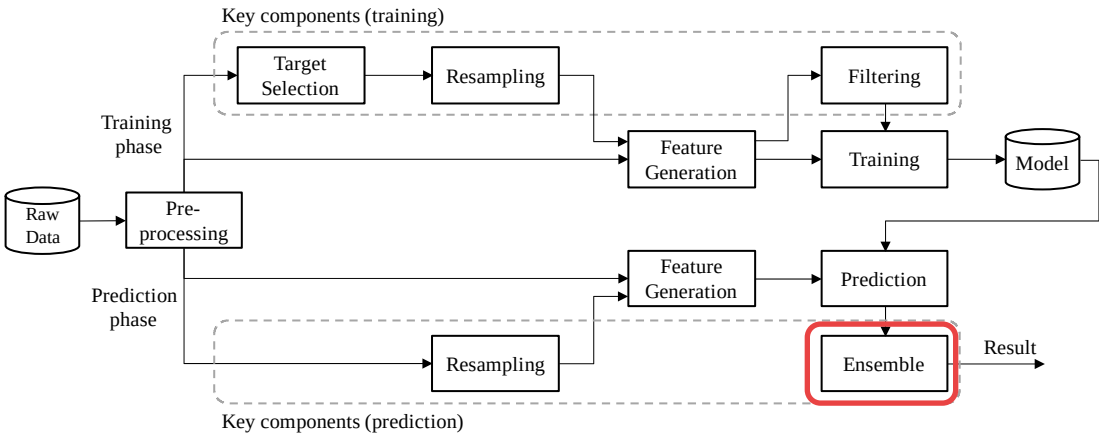
Proposal: Filtering

- Distance-based strategy to filter pseudo vectors
  - For each target, predict scores of pseudo feature vectors
  - Drop pseudo ones which have large gap with the original
    - Thresh = average + f * standard deviation
    - f is a control parameter



Proposal: Ensemble

- Different with synthetic data, regenerated vectors are linked to the original one
- Ensemble scores of multiple feature vectors for the same target can obtain better representation
  - The simplest strategy is to take the average



Evaluation: Overview

Data and features sets

- **Three major types of traffic data (Network flows, HTTP log, PDNS)**
- **Three well cited features sets**

X

Classifier

- **Three major classifiers: Random Forest, DNN and SVM**

Evaluation: Conventional methods

- Original
  - without any extra data
- Noise
  - adding uniform noise to each feature vectors
- SMOTE
  - using SMOTE to increase the number of malicious training sets to the same number of the benign class (the neighborhood number is 5)

Evaluation: Patterns of proposal

Pattern	Resample in training	Resample in prediction	Target selection & filtering
Alchemy 1	✓		
Alchemy 2		✓	
Alchemy 3	✓	✓	
Alchemy 4	✓	✓	✓

Evaluation: Parameters

	Network flows	Passive DNS	HTTP logs
resampling iterations	20	20	20
resampling option	#1	#4	#4
$[P_{lower}, P_{upper}]$	[0, 0.98]	[0, 0.999]	[0, 0.999]
f for filtering	5	5	5

Evaluation: Environment

CPU	XEON 2.7 GHz 8 cores * 2
Memory (RAM)	256 GB
OS / Language	Ubuntu 16.04 / Python 3.6
Library	NumPy, Pandas, scikit-learn, TensorFlow

Evaluation: Datasets

NTT

Network flows

class	# of flow	# of IP address	# of valid flow	# of valid IP address
Benign	14,743,140	2,431	14,737,520	1,984
Malicious	261,610	455	258,270	219

HTTP log

class	# of log	# of IP address	# of valid log	# of valid IP address
benign	1,708,120	2,251	1,706,532	1,487
malicious	5,096	113	5,058	95

PDNS

class	# of record	# of do-main	# of valid record	# of valid domain
Benign	3,333,877	4,834	3,333,877	4,834
Malicious	12,089	329	10,919	279

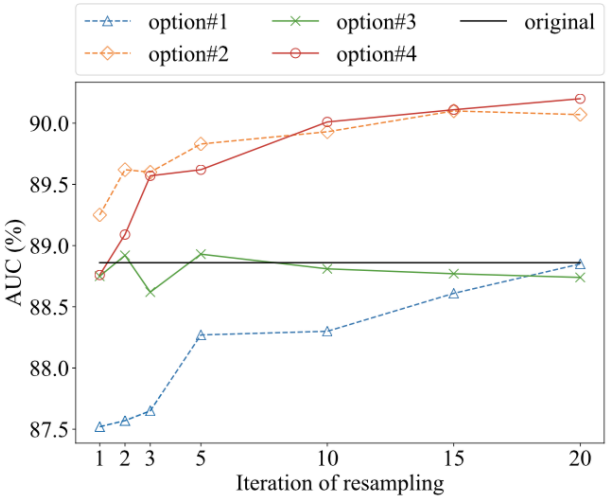
Evaluation: AUC Comparison

- Alchemy outperformed conventional methods in all types of data * classifiers

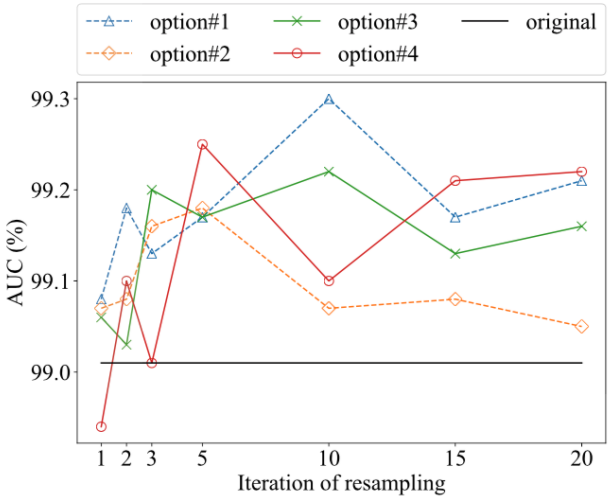
Target	AUC (%)								
	Network Flows			Passive DNS			HTTP logs		
	RF	DNN	SVM	RF	DNN	SVM	RF	DNN	SVM
Original	88.61	80.33	75.88	99.03	97.54	97.54	93.15	83.05	82.99
Noise	88.71	80.80	78.21	98.81	97.69	97.59	92.61	84.05	83.49
SMOTE	89.26	80.81	75.24	99.13	97.06	97.89	93.67	83.66	87.57
Alchemy 1	88.70	83.94	80.07	99.18	97.54	97.39	94.34	87.40	87.15
Alchemy 2	83.60	75.65	75.52	99.13	97.56	97.98	92.98	87.39	83.11
Alchemy 3	**90.07**	**85.21**	**82.07**	**99.19**	**97.82**	**98.66**	**94.58**	**88.96**	**88.38**
Alchemy 4	**90.13**	**85.00**	**82.06**	**99.21**	**97.84**	**98.59**	**94.76**	**89.17**	**88.19**

Evaluation: Resampling V.S. AUC

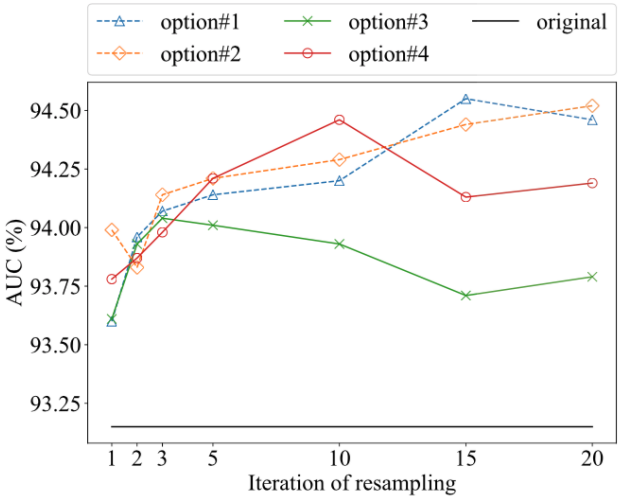
- When iterations of sampling are increased, AUC scores are improved



(a) Network flow



(b) Passive DNS



(c) HTTP log

Limitations

- Frequency may be lost due to random sampling
  - Resample continuous N flows
- Features independent from traffic bag / sampling (e.g., domain string for a target domain)
  - Review variance/invariance of features sets before applying Alchemy

Conclusion

- We propose a method (Alchemy) to resample several subsets of raw traffic and regenerate multiple pseudo feature vectors to represent each host, regardless of data, features and classifiers
- Alchemy outperformed conventional methods in all types of data * classifiers

Reference

1. M.Antonakakis,T.April,M.Bailey,M.Bernhard,E.Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Men- scher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Under- standing the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 1093–1110, 2017.
2. L.Bilge,D.Balzarotti,W.Robertson,E.Kirda,andC.Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 129– 138, ACM, 2012.
3. L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive DNS analysis.," in *NDSS*, 2011.
4. K. Bartos, M. Sofka, and V. Franc, "Optimized invariant repre- sentation of network traffic for detecting unseen malware vari- ants.," in *USENIX Security Symposium*, pp. 807–822, 2016.