New Directions in Automated Traffic Analysis

Jordan Holland, Paul Schmitt, Nick Feamster*, Prateek Mittal
Princeton University, University of Chicago*
ACM CCS '21

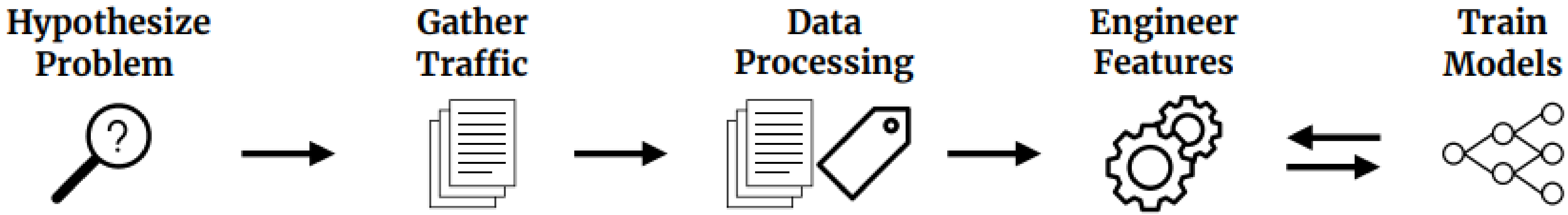
GyeongHeon Jeong(ghjeong@mmlab.snu.ac.kr)

Index

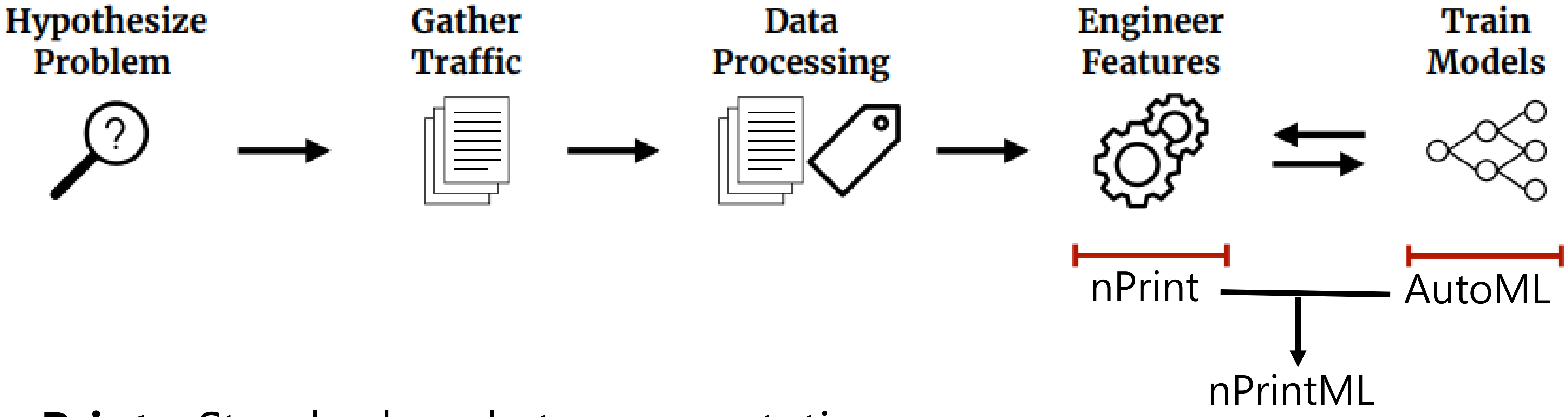
Introduction

- Many traffic analysis tasks in network security rely on machine learning
  - Application Identification, Device Fingerprint, OS Detection, Anomaly Detection, …

- Classic ML Pipeline



Introduction



- **nPrint** : Standard packet representation
  - Encoding each packet in inherently normalized, binary representation while preserving the underlying semantics of each packet.
- **nPrintML** : nPrint + AutoML
  - AutoML : existing automated machine learning tool
  - Enabling automated model selection and hyperparameter tuning

nPrint – Design Requirement

TCP Segment Header Format

Bit #	0 7	8 15	16 23	24 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset Res	Flags	Window Size	
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 7	8 15	16 23	24 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

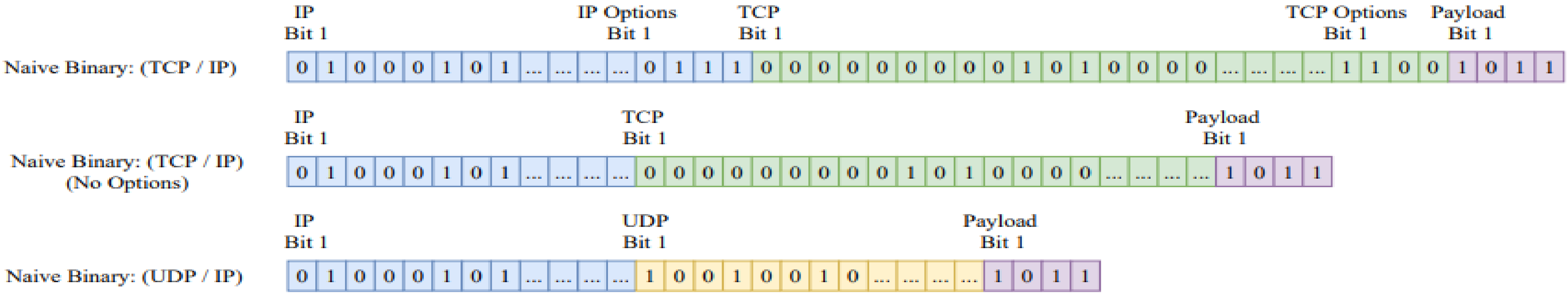
TCP & UDP Header

nPrint – Design Requirement

- Complete
  - Including every bit of a packet header
- Constant size per problem
  - Many machine learning models assume that inputs are always the same size
- Normalized
  - Machine learning models typically perform better when features are normalized
- Aligned
  - Every location in the representation should correspond to the same part of the packet header

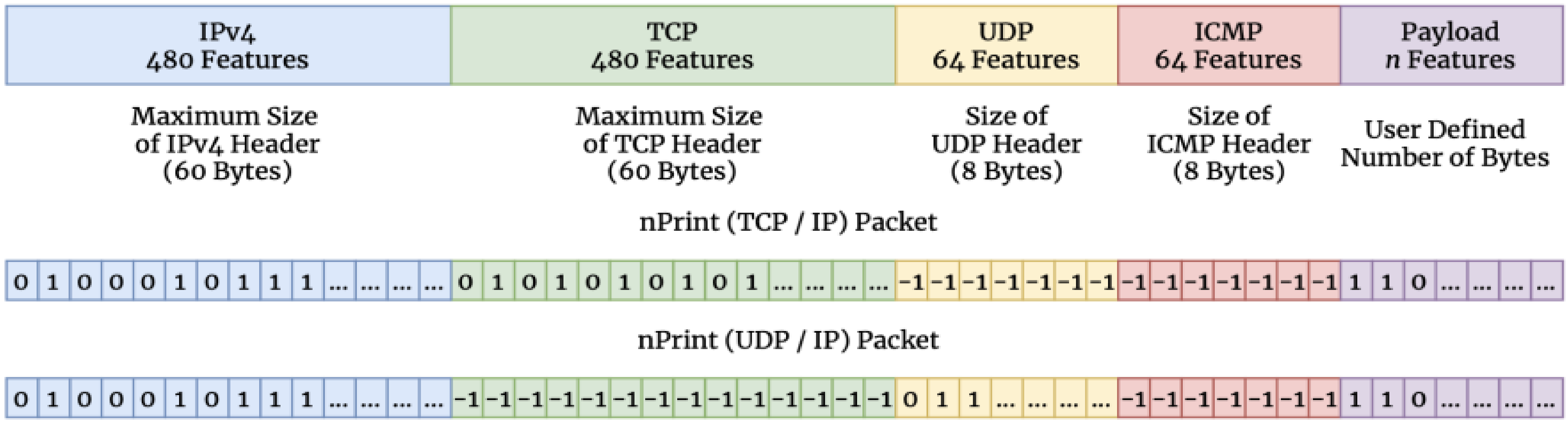
nPrint – Building Standard Data Representation

- Semantic Representation
  - Collecting IP TTL, TCP port number, UDP length, …
  - It needs expertise to parse semantic structure of every protocol, determining the correct representation of each feature needs effort
- Naive Binary Representation
  - Because it is misaligned, two packets have different meanings for the same feature



nPrint – Building Standard Data Representation

- nPrint
  - Hybrid of semantic and binary packet representations
  - Filling non-existing headers with -1 (internal padding)
  - Enable to understand the features that are driving the performance of model with mapping to semantic structure because it is aligned



nPrint – Implementation

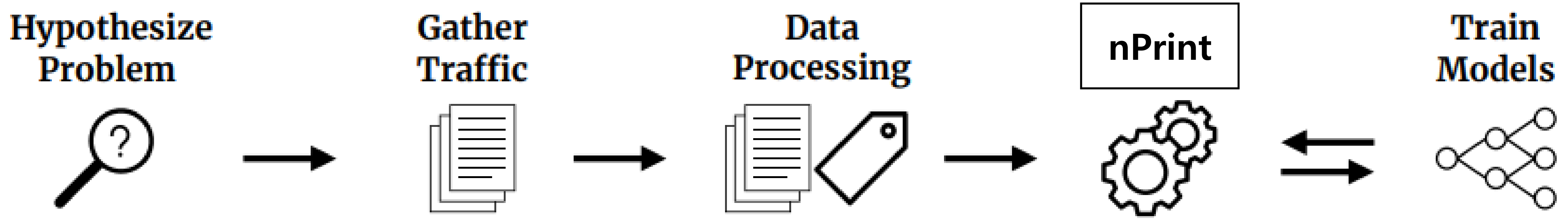
- nPrint transforms over 1.5 million packets per minute.
  - CSV output, libpcap for packet processing

- nPrint has a constant memory footprint.

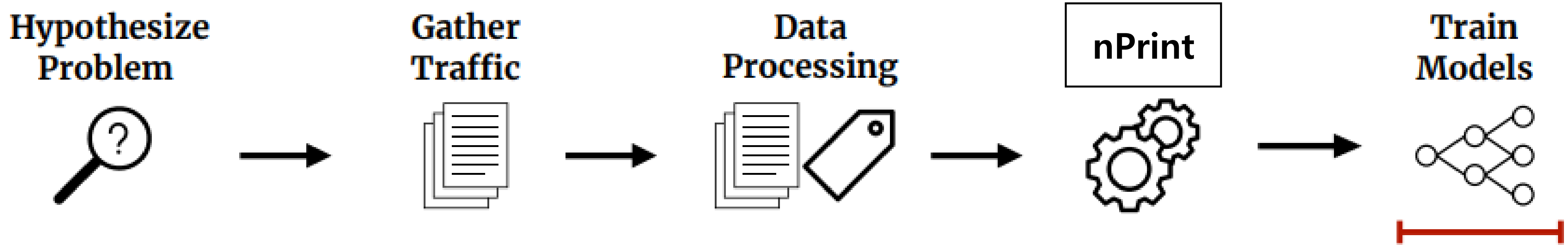
- Proof of Concept
  - Amenable to parallelization
  - 16 process & 8Gbps live traffic load with near zero loss

nPrint

- nPrint replace Feature Engineering



- Next step is automating of Model Training with AutoML

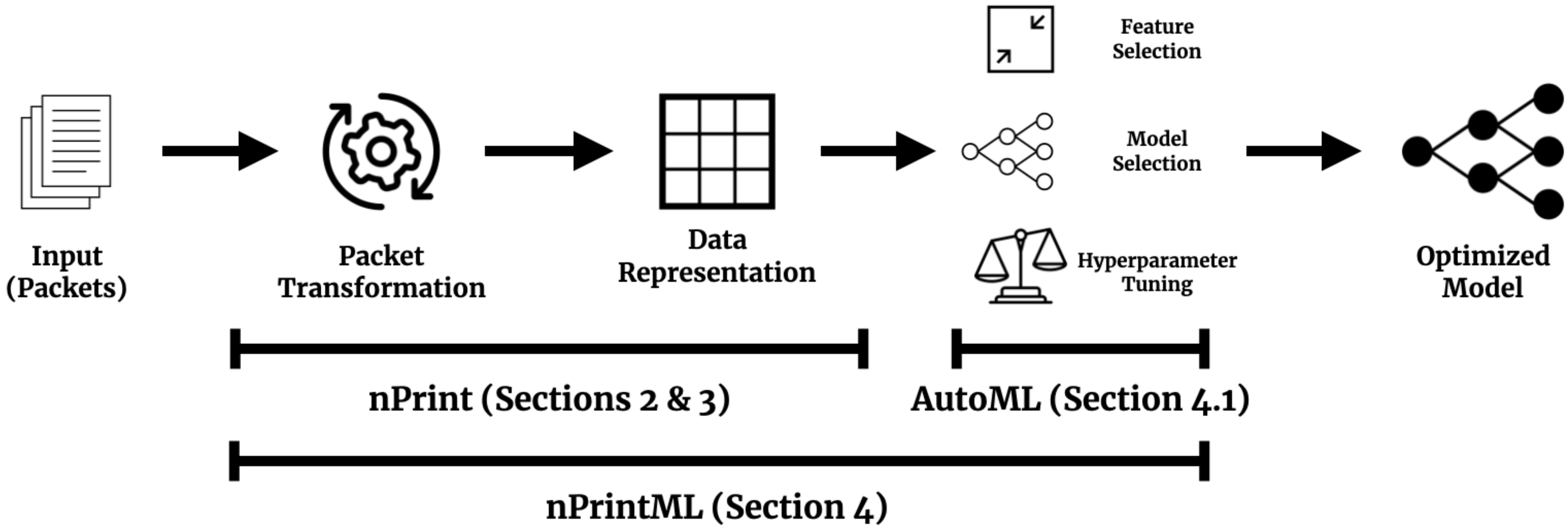


nPrintML - AutoGluon AutoML

- AutoML : Tools designed to automate feature selection, model selection, and hyperparameter tuning to find an optimized model
  - Not only just one model, but all model we use
    1) we can train and test more model types
    2) we can optimize the hyperparameters for every model we train
    3) we are certain that the best model is chosen for a given representation

- AutoGluon : AutoML tool which is open-source project in Amazon
  - Model ensembling achieves higher performance than other AutoML tools
  - Train models from 6 base classes
    - Random forests, DNN, KNN, …
  - No limit on training time, allowing to find the best model

nPrintML



Detailed Traffic Analysis with nPrintML

Case Study - Active Device Fingerprinting

- Active Device Fingerprinting : Identification of traffic`s device with sending probe
- Nmap (Network Mapping)
  - Well known and used device fingerprinting
  - Over 20 years of hand curated features and hand-developed heuristic to fingerprint devices
  - Additionally, run AutoML
- nPrintML
  - Make nPrint representation automatically with same input packet
  - Run AutoML
- nPrintML is better than Nmap even if it take benefit of automation

Vendor	Average Precision	
	ML-Enhanced Nmap	nPrint
Adtran	1.00	1.00
Avtech	0.87	0.95
Axis	0.93	0.98
Chromecast	1.00	1.00
Cisco	0.97	0.99
Dell	0.85	0.99
H3C	0.95	0.96
Huawei	0.94	0.95
Juniper	0.99	0.99
Lancom	0.99	0.99
Mikrotik	0.88	0.91
NEC	1.00	1.00
Roku	0.92	0.99
Ubiquoss	0.99	0.99
ZTE	0.99	0.99

Case Study - Passive OS Fingerprinting

- Passive OS Fingerprinting : Identification of OS behind any traffic without sending probe

- p0f (Passive OS Fingerprinting)
  - Depend on user-curated signature database
  - Looks for direct matches in its database in order to identify the OS

- Input packet : CICIDS2017 intrusion detection evaluation dataset
  - Use 100,000 packets for each device, and split them to 1, 10, 100 PCAPs (Packet CAPture)

- nPrintML outperforms p0f

Case Study - Passive OS Fingerprinting

		p0f						nPrint					
		1 Packet		10 Packets		100 Packets		1 Packet		10 Packets		100 Packets	
Host	p0f Label	P	R	P	R	P	R	P	R	P	R	P	R
Mac OS	Mac OS x 10.x	1.00	0.05	1.00	0.28	1.00	0.88	0.99	0.99	1.00	0.99	1.00	0.99
Web Server	Linux 3.11 and newer	1.00	0.01	1.00	0.25	1.00	0.74	0.99	0.99	0.99	1.00	0.99	1.00
Ubuntu 14.4 32B		1.00	0.04	1.00	0.20	1.00	0.69						
Ubuntu 14.4 64B		1.00	0.04	1.00	0.20	1.00	0.65						
Ubuntu 16.4 32B		1.00	0.05	1.00	0.19	1.00	0.68						
Ubuntu 16.4 64B		1.00	0.04	1.00	0.24	1.00	0.79						
Ubuntu Server		1.00	0.05	1.00	0.25	1.00	0.74						
Windows 10	Windows 7 or 8	0.99	0.00	0.98	0.02	0.98	0.09	1.00	1.00	1.00	1.00	1.00	1.00
Windows 10 Pro		0.99	0.01	0.98	0.04	1.00	0.14						
Windows 7 Pro		1.00	0.04	1.00	0.23	1.00	0.71						
Windows 8.1		0.99	0.05	0.99	0.25	0.99	0.77						
Windows Vista		1.00	0.01	1.00	0.27	1.00	0.71						
Kali Linux	No output	-	-	-	-	-	-	-	-	-	-	-	-

*precision : TP/(TP+FP) , recall : TP/(TP+FN)

Case Study - Passive OS Fingerprinting

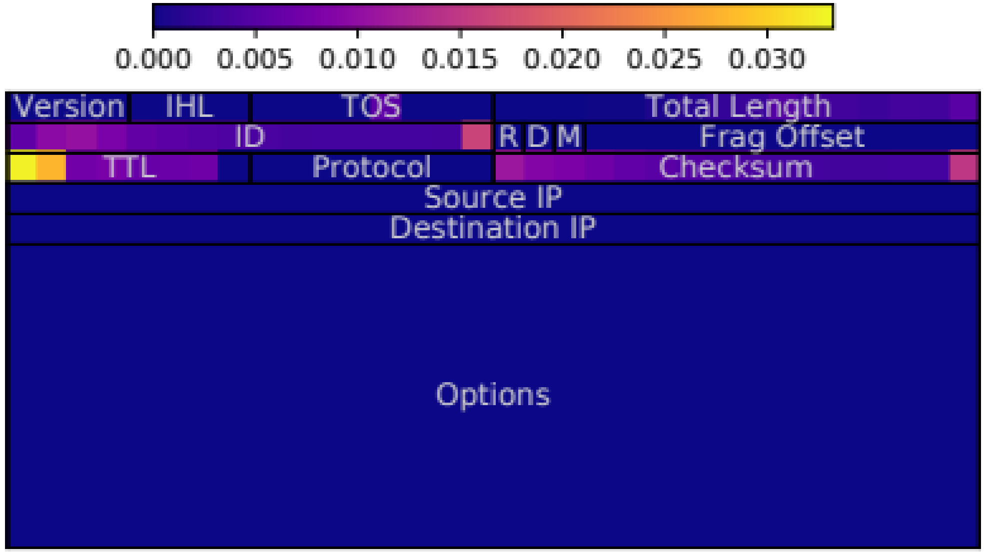


Figure A. Feature importance heatmap (IPv4)

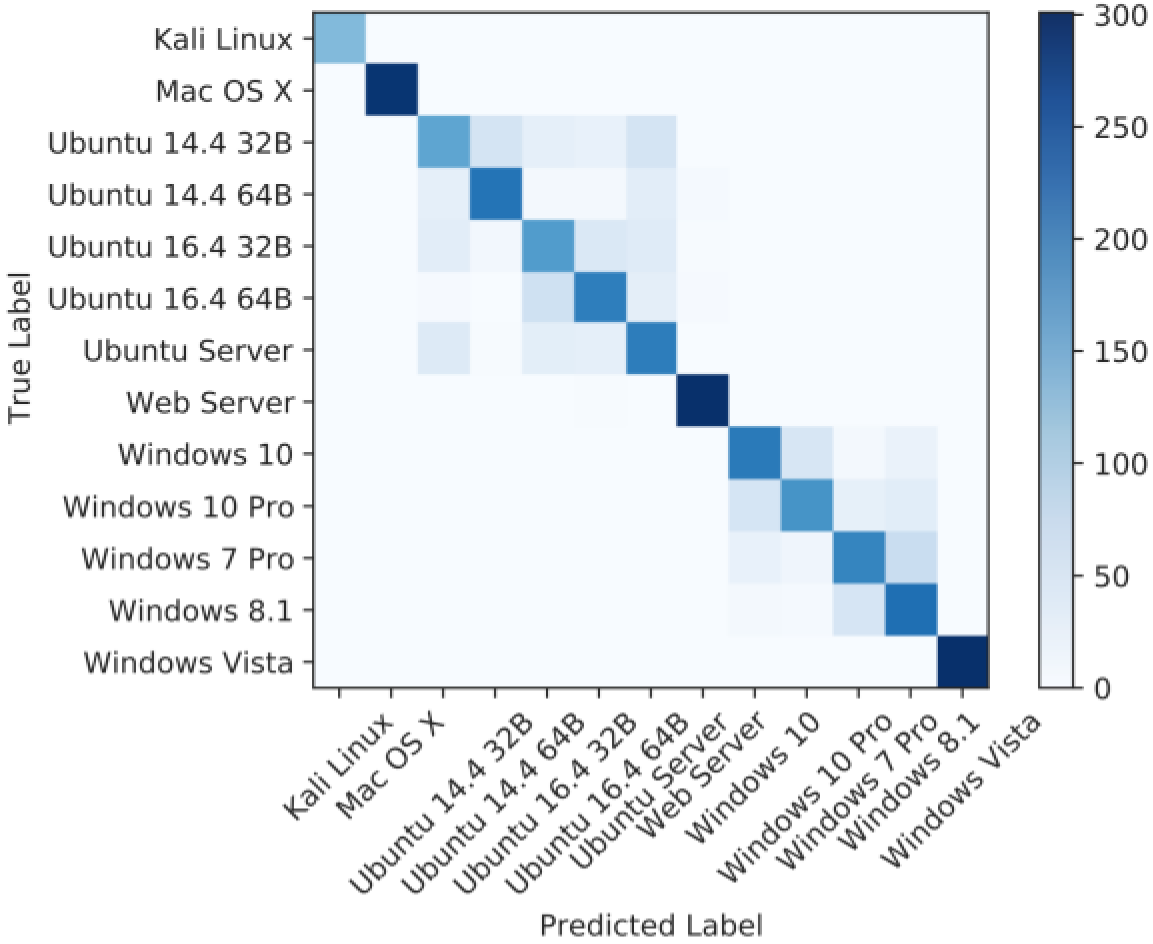


Figure B. nPrintML confusion matrix

Case Study - DTLS Application Identification

- DTLS Application Identification : Identify application and browser that generated DTLS handshake
- Input packet : 7,000 DTLS handshake traffic with 7 classes

- nPrintML
  - Can automatically detect features in noisy environment.
  - Performs well across models and trains quickly.

Model Architecture	Fit Time (Seconds)	Total Inference Time (Seconds)	F1
Random Forest	3.69	0.37	99.8
ExtraTrees	3.89	0.43	99.9
KNeighbors	3.90	8.95	96.0
LightGBM	5.21	0.15	99.8
Catboost	9.00	0.38	99.7
Weighted Ensemble	46.1	0.45	99.9
Neural Network	85.58	29.9	99.7

*F1 score = $\frac{2}{precision^{-1}+recall^{-1}}$ * 100

Case Study – Additional Case Studies

- netML Challenge Examples
  - Malware detection for IoT devices, intrusion detection, and traffic identification

Problem Overview			**nPrintML**					**Comparison**	
Description	Dataset	# Classes	Configuration eAppendix A.4)	Sample Size (# Packets)	Balanced Accuracy	ROC AUC	Macro F1	Score	Source
Malware Detection for IoT Traces (§5.4.1)	netML IoT [6, 28]	2	-4 -t -u	10	92.4	99.5	93.2	99.9 (True Positive Rate)	NetML Challenge Leaderboard [37]
		19			86.1	96.9	84.1	39.7 (Balanced F1)	
Type of Traffic in Capture (§5.4.1)	netML Non-VPN [6, 12]	7	-4 -t -u -p 10	10	81.9	98.0	79.5	67.3 (Balanced F1)	
			-4 -t -u		76.1	94.2	75.8		
		18			66.2	91.3	63.7	42.1 (Balanced F1)	
		31			60.9	92.2	57.6	34.9 (Balanced F1)	
Intrusion Detection (§5.4.1)	netML CICIDS 2017 [6, 48]	2	-4 -t -u	5	99.9	99.9	99.9	98.9 (True Positive Rate)	
		8			99.9	99.9	99.9	99.2 (Balanced F1)	

* Balanced Accuracy = (Sensitivity + Specificity) / 2 ** (Sensitivity = TP/(TP+FN) , Specificity = TN/(TN+FP))
* ROC AUC (Receiver Operating Characteristic Area Under Curve)

Case Study – Additional Case Studies

- Mobile Country of Origin
  - Use Cross-platform dataset to determine country of origin of mobile application traces
- Streaming Video Providers
  - Tested whether video services can be identified through video traffic analysis
  - Each streaming video service player may exhibit individualistic flow behavior to deliver video traffic

Problem Overview			**nPrintML**					**Comparison**	
Description	Dataset	# Classes	Configuration eAppendix A.4)	Sample Size (# Packets)	Balanced Accuracy	ROC AUC	Macro F1	Score	Source
Determine Country of Origin for Android & iOS Application Traces (§5.4.2)	Cross Platform [44]	3	-4 -t -u -p 50	25	96.8	90.2	90.4	No Previous Work	
Identify streaming video (DASH) service via device SYN packets (§5.4.3)	Streaming Video Providers [10]	4	-4 -t -u -R	10	77.9	96.0	78.9	No Previous Work	
				25	90.2	98.6	90.4		
				50	98.4	99.9	98.6		

Conclusions

- New direction of automatic traffic analysis
- Standard packet representation, nPrint, automates parts of ML process
- nPrintML optimize models for each task by training feature selection, model selection, and hyperparameter tuning

Critiques

- nPrintML has other problems such as automated timeseries analysis and classification involving multiple flows

- Representing packet in nPrint format, packet become much bigger than previous one, then it can cause overhead
  - seems solution must be needed

Thank you for listening