TempoCode-IoT: temporal codebook-based encoding of flow features for intrusion detection in Internet of Things

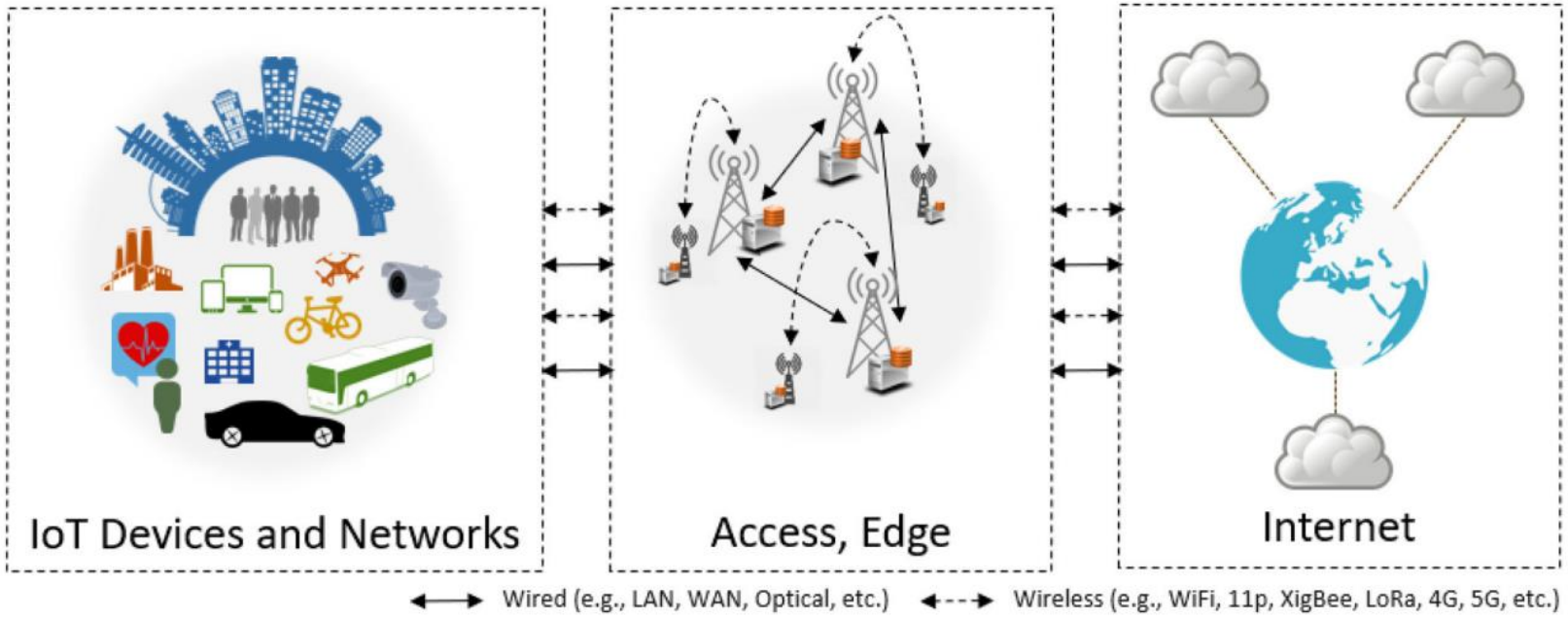
Abdul Jabbar Siddiqui and Azzedine Boukerche
University of Ottawa, Canada

Cluster Computing 2021

Outline

- Introduction
- Design
  - Overview
  - Feature extraction
  - Codebook generation
  - Learning and testing
- Evaluation
- Conclusion

Introduction



- Growing plethora of connected devices and things are reshaping the landscape of an Internet-of-Things (IoT)
- Due to the increasing diversity, novel threats and attack vectors are suggested

* Blockchain Signaling System (BloSS): Cooperative Signaling of Distributed Denial-of-Service Attacks, Journal of Network and Systems Management, 2020

Importance

- Threats of IoT devices are harmful to not only device itself, but also human life and properties

- Attacks on smartcar, e-health devices are directly connected to human life

- Attacks on smartcity infra, bank lead to losses of properties

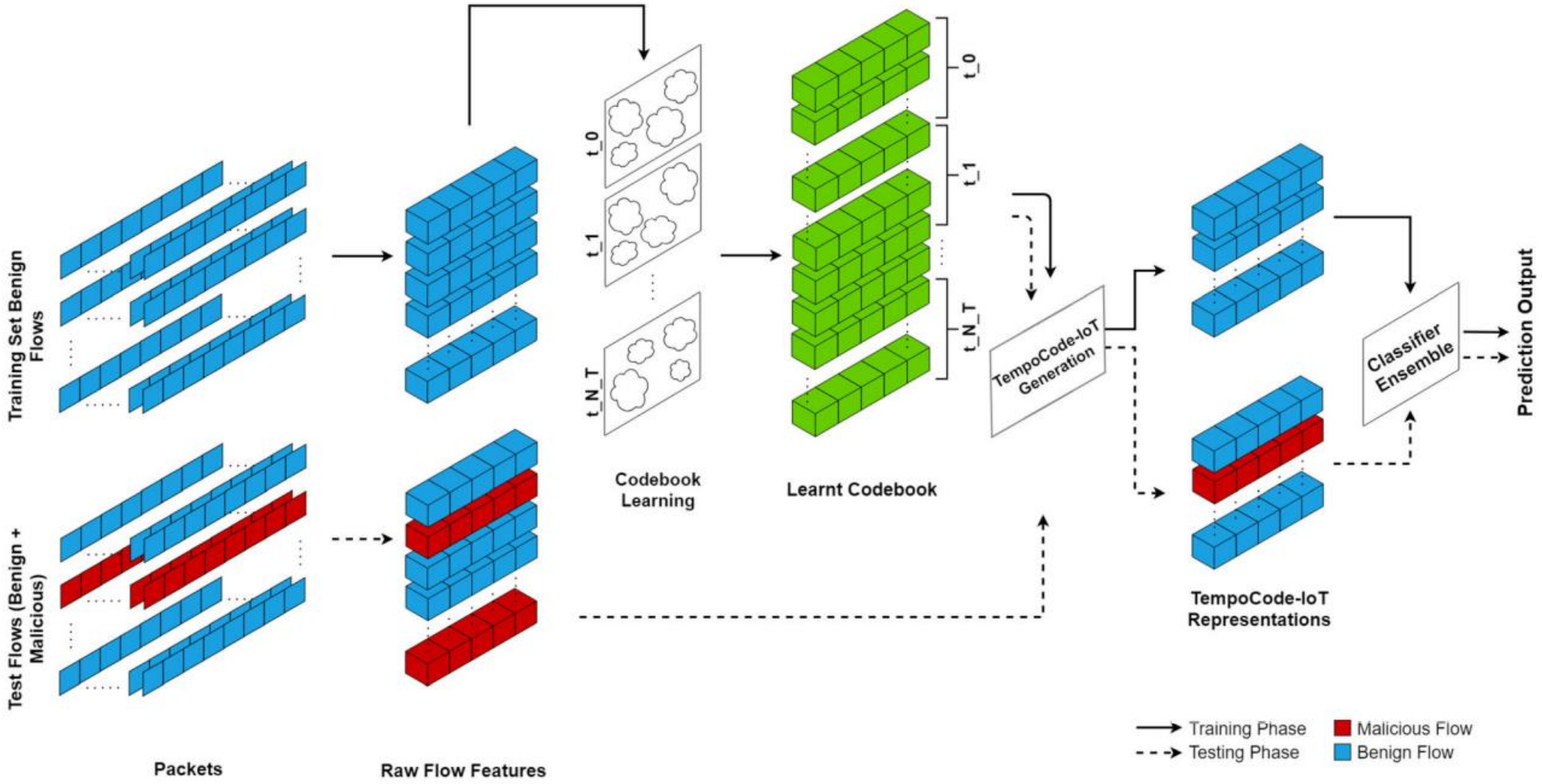
➔ Intrusion detection on IoT devices becomes more important

Codebook-based IDS

- Transforming flow-based features into more discriminative representations
- Designing an ensemble of classifiers based on these to differentiate between benign and malicious flows
- The method is designed to serve in a centralized IDS, leveraging the compute and storage resources therein

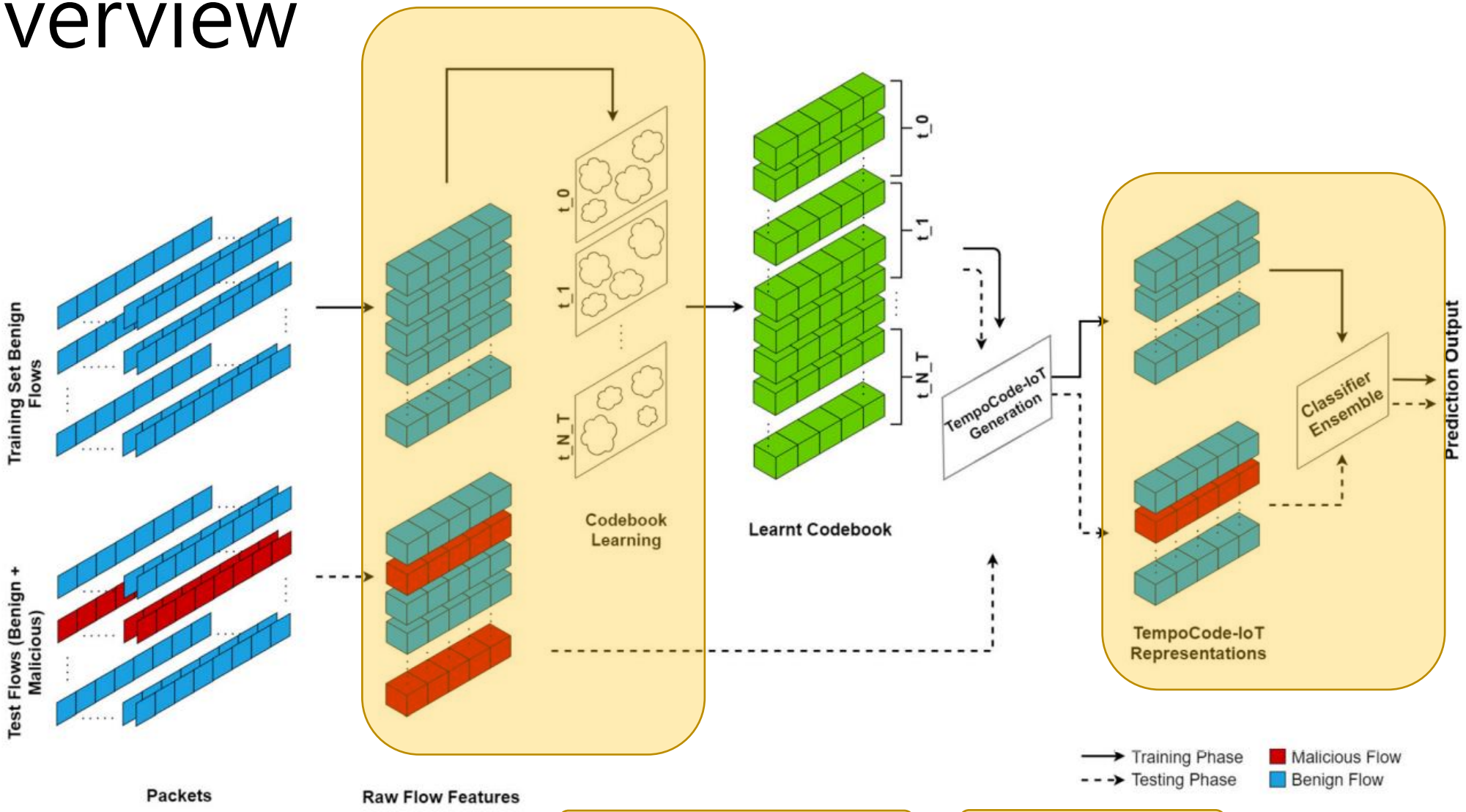
Design

Overview



Packet capture → Feature extraction → Codebook generation → Training/Testing → Classifying

Overview



Packet capture → Feature extraction → Codebook generation → Training/Testing → Classifying

Overview

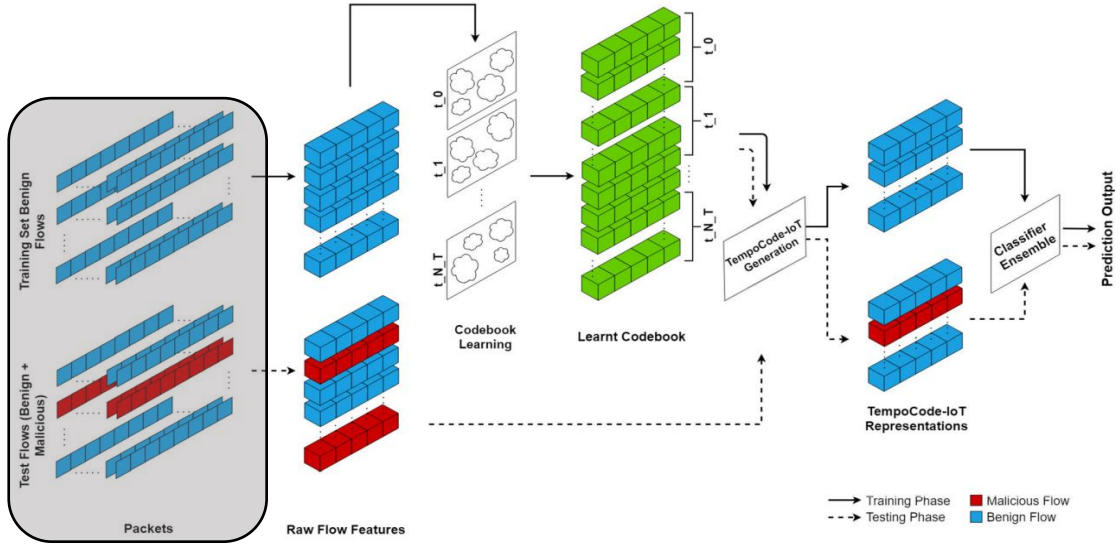
- Process of this work is common
  - Widely used concepts of feature extraction, machine learning and classifying

- Characteristics of this work are in “codebook generation”
  - Somewhat in ensemble learning

Packet capture

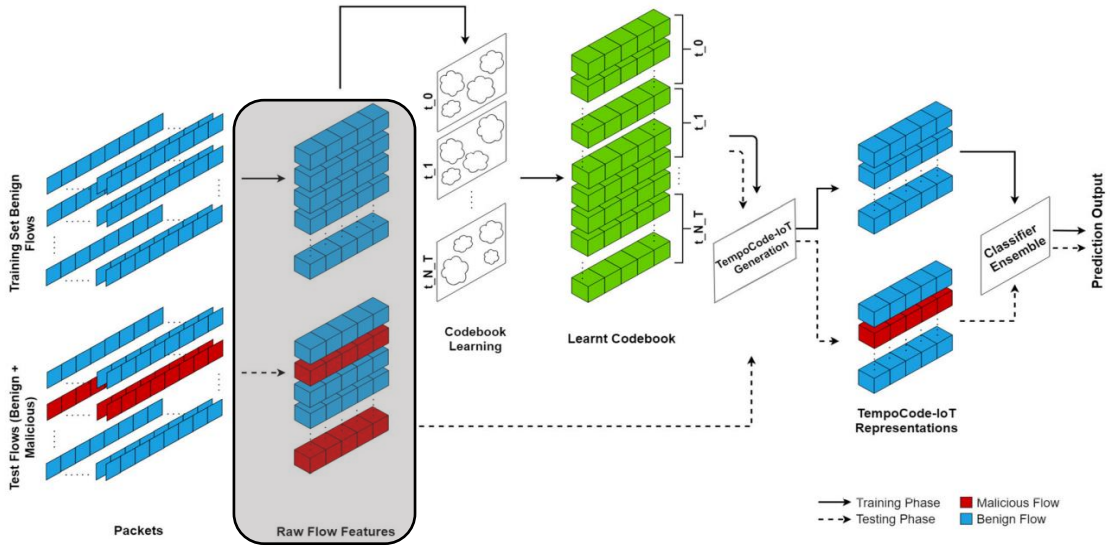
- This work uses public dataset only
  - No self-captured packet dataset

- NBaIoT and CICIDS2017 datasets are used



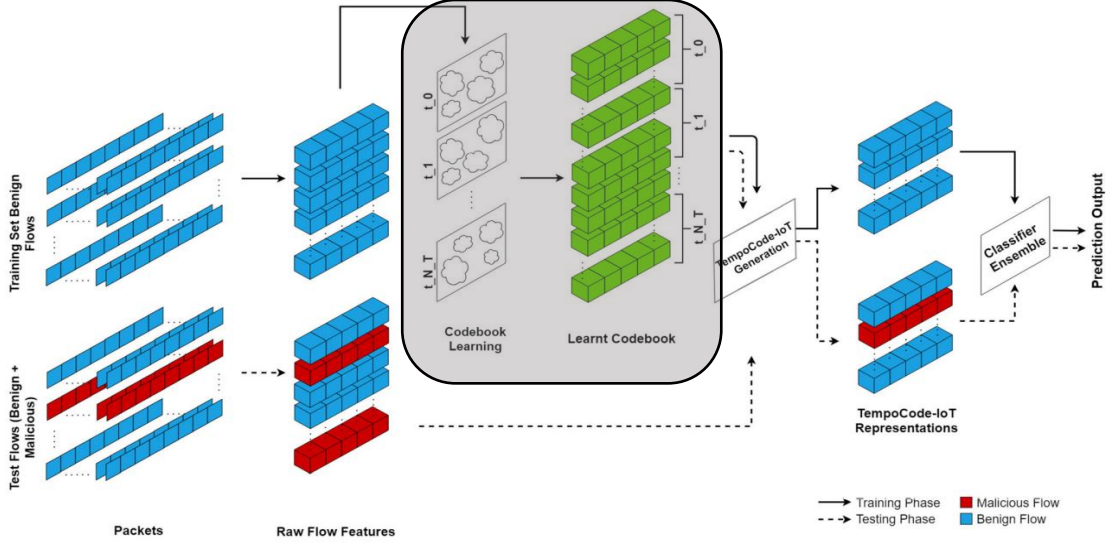
Feature extraction

- Features are already extracted (or selected) in dataset
  - Authors just utilized them



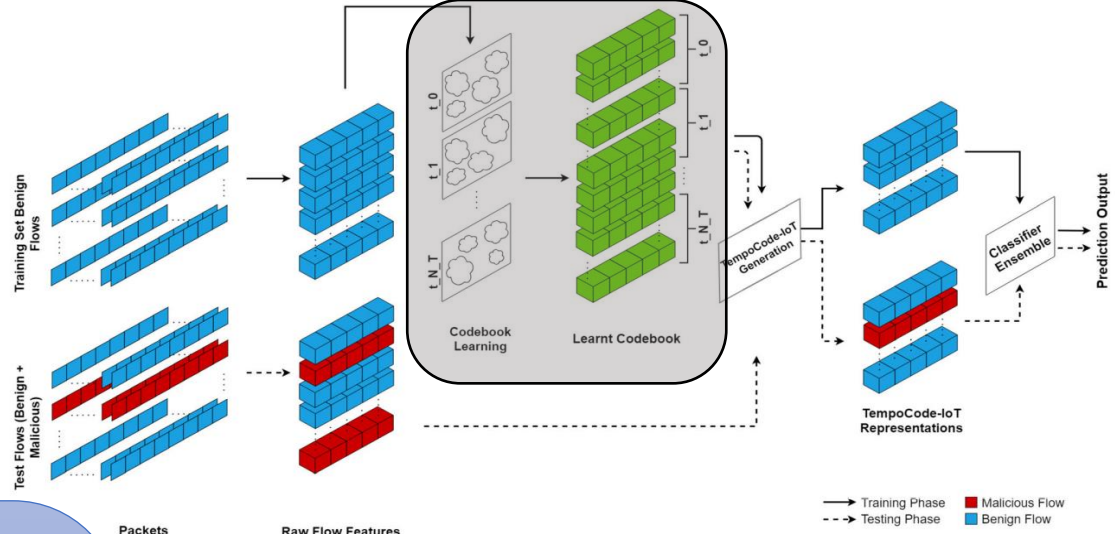
- Features examples
  - Flow Duration, packet Length (min, mean, stdev in forward and backward direction), subflow bytes (in forward direction), flow inter-arrival time (min, mean, stdev in forward and backward direction), active_min, active_mean

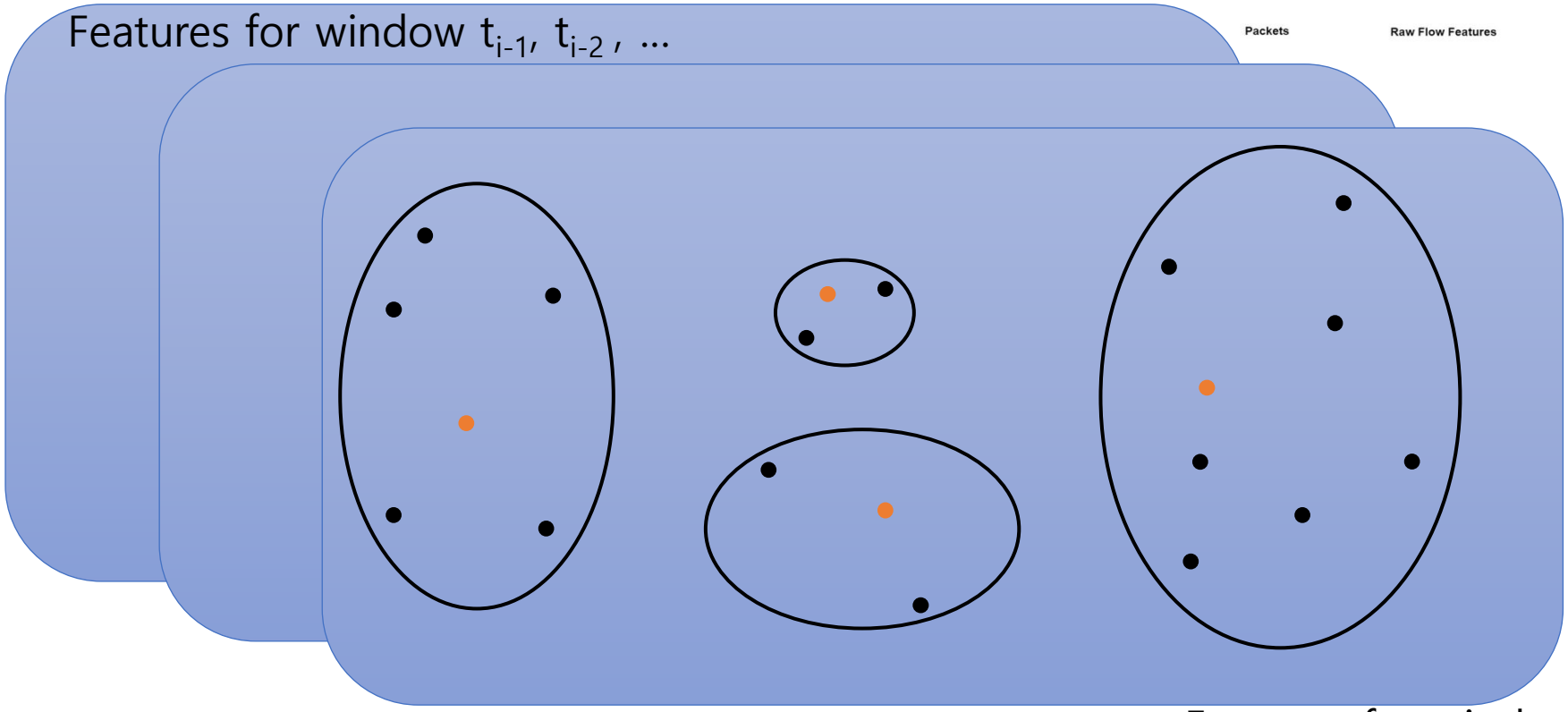
Codebook generation



- Only benign flows are used to generate codebook
- In each time window t_i, a clustering method such as KMeans is applied to learn N_{ct} key patterns as codewords to represent the benign traffic in t_i
  - N_{ct} : the number of codewords (the number of clusters)
- In short, they run nearest neighbor for all features in each window, centers become the codewords

Codebook generation



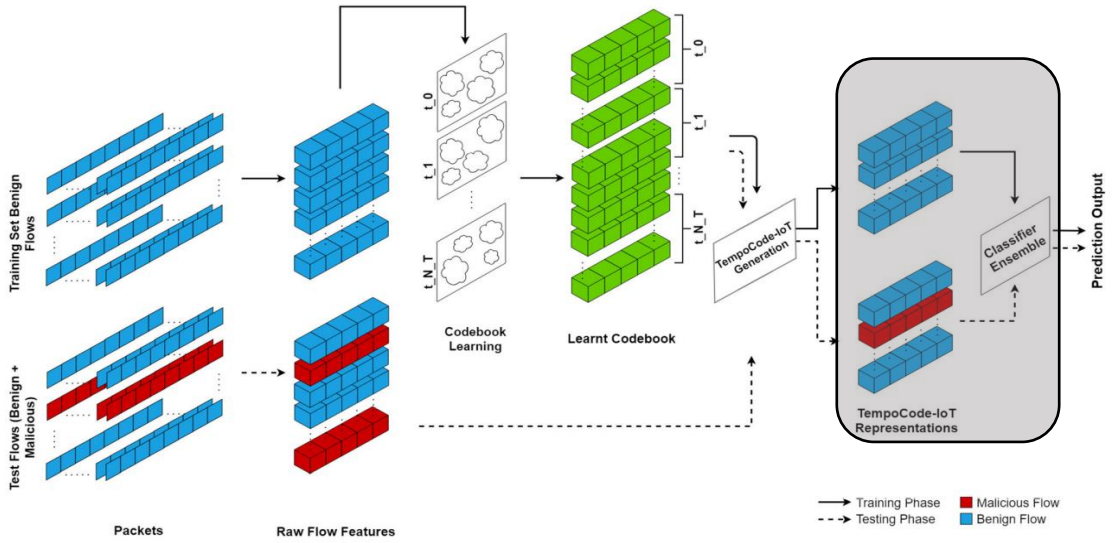


- data point (feature)
- codeword (center)

Features for window t_i , N_{ct}=4

Ensemble training

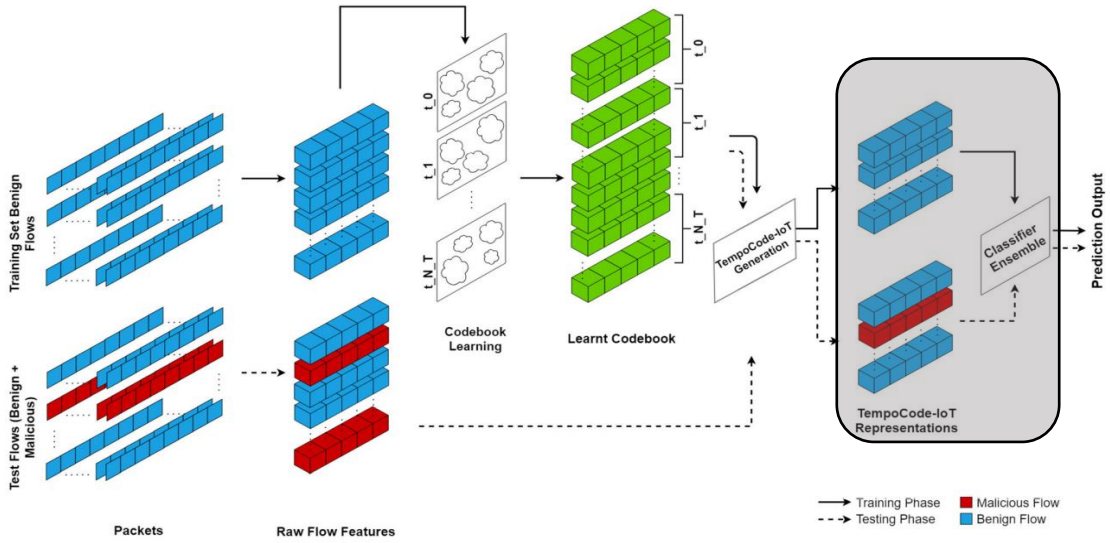
- Each classifier in ensemble is trained on a random subset of the training dataset



- The prediction of each constituent classifier is then combined through voting to produce the overall classification output
  - Detailed procedure of voting is omitted in the paper

- Pros: Parallelism & Better generalization ability

Classifying



- Scoring is based on distance

$$TC_j = \{q_{t_i,k} | i = 0, \ldots, N_T; k = 1, \ldots, N_{ct}\}$$

  - bin ($q_{t_i,k}$) holds the distance of F_j from $cw_{t_i,k}$

- SVM is chosen as base classifier
  - A single multi-class SVM classifier is built by collecting many such binary classifiers, depending on the number of classes in the dataset

Evaluation

Dataset description – CICID2017

Table 2 Attack types in CICIDS2017 datasets

Attack type	Description
Bruteforce	Based on the FTP- and SSH-Patator tools. The attacker tries to gain access to content or documents via a hit and try method
Heartbleed	Targeted against OpenSSL-based Transport Layer Security (TLS) protocol
Botnet	A number of devices are compromised and exploited to carry out different attacks/operations. Ares-based Botnet
DoS/DDoS	Targeted against a network resource or service to make it unavailable for benign users. When many different devices are exploited (e.g. by a botnet), it is called DDoS. Tools used: GoldenEye, Slowloris, Hulk, Slowhttptest, Heartleech, LOIC
Web attack	Attacks like SQL Injection or Cross-Site Scripting (XSS), over the web, exploiting vulnerabilities in code
Infiltration attack	Internally originated attacks. Attacker exploits software vulnerabilities to setup a backdoor on victim devices to carry out various attacks such as portscan or IP sweep, etc. Tool: Metasploit, Nmap, portscan

Dataset description - NBaIoT

Table 4 Attack types in NBaIoT dataset

Attack type	Botnet family	Description
Scan	Mirai and BASHLITE (Gafgyt)	Looks for vulnerable devices in the network
Junk	BASHLITE (Gafgyt)	Sends junk/spam data
COMBO	BASHLITE (Gafgyt)	Sends spam data; Opens connection to a given IP address and port
Flooding	Mirai	ACK, SYN, UDP, UDPplain (higher PPS, enabled by fewer options)
	BASHLITE (Gafgyt)	UDP, TCP

ID	Device	#Benign	#Mirai	#Gafgyt
1	Danmini (doorbell)	40395	652100	316650
2	Ecobee (thermostat)	13111	512133	310630
3	Ennio (doorbell)	34692	N/A	316400
4	Philips B120N10 (baby monitor)	160137	610714	312723
5	Provision PT737E (security camera)	55169	436010	330096
6	Provision PT838 (security camera)	91555	429337	309040
7	Samsung SNH1011N (Webcam)	46817	N/A	323072
8	SimpleHome XC57-1002-WHT (security camera)	42784	513248	303223
9	SimpleHome XC57-1003-WHT (security camera)	17936	514860	316438

Effect of time window

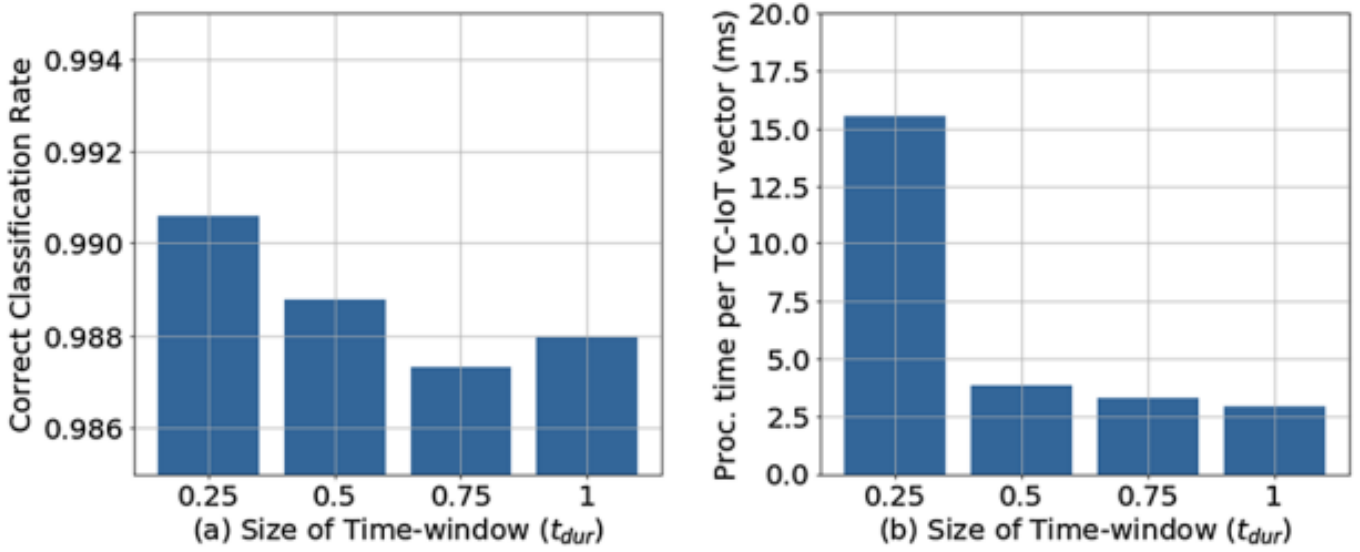


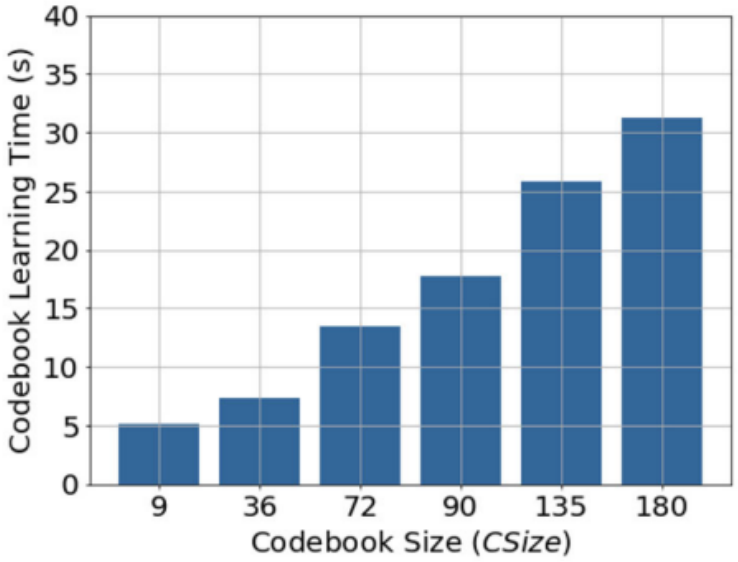
Fig. 3 **a** Effect of t_{dur} on accuracy (correct classification rate of benign and malicious samples) and **b** processing time (per TemPoCode-IoT vector)

- Shorter time window, higher accuracy, longer processing time
- 0.75 shows, however, there can be “overfitting”

Effect of # of centers (in codebook)

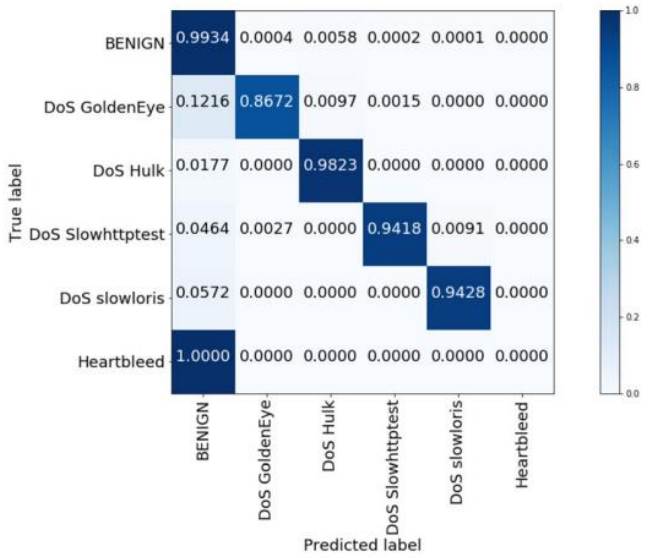
Table 6 Effect of N_{ct} and *CSize* (codebook size) on TempoCode-IoT classification scores

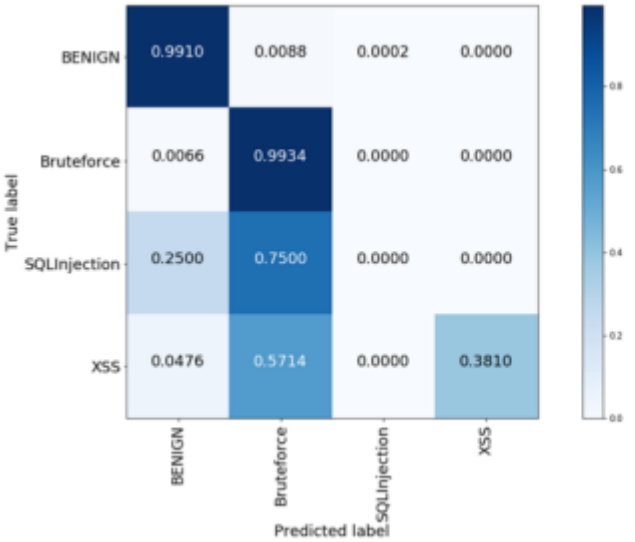
N_{ct}	CSize	Ben-Ben	Ben-Mal	Mal-Ben	Mal-Mal	Prec-Ben	Recall-Ben	F1-Ben	Prec-Mal	Recall-Mal	F1-Mal
5	45	345555	2813	2917	108394	0.9916	0.9920	0.9918	0.9747	0.9738	0.9743
10	90	346495	1873	2577	108734	0.9926	0.9946	0.9936	0.9831	0.9769	0.9800
15	135	346618	1750	**2534**	**108777**	**0.9927**	0.9950	**0.9939**	0.9842	**0.9772**	**0.9807**
20	180	**346762**	**1606**	2794	108517	0.9920	**0.9954**	0.9937	**0.9854**	0.9749	0.9801

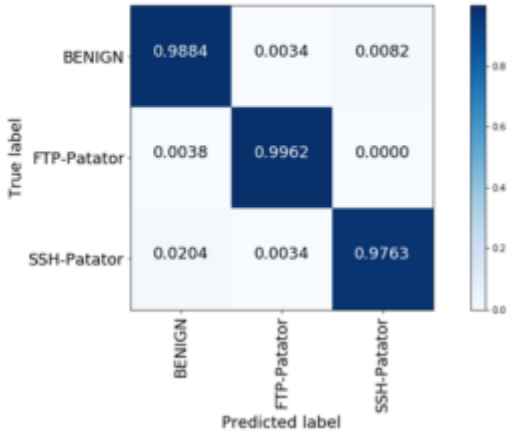


- More codewords, higher accuracy, longer learning time
- 15 is the best, however, it shows that parameter can change depending on dataset

Performance results

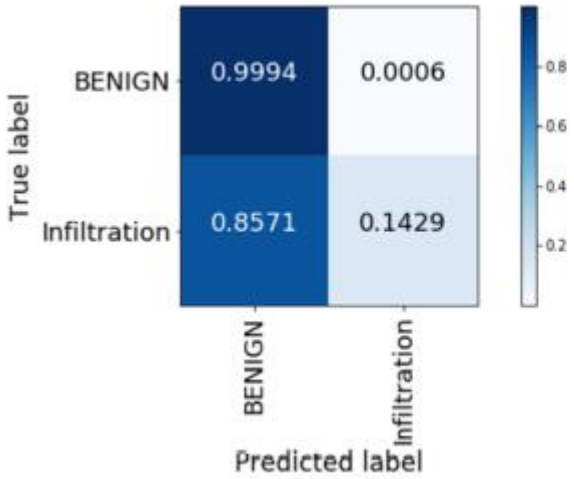


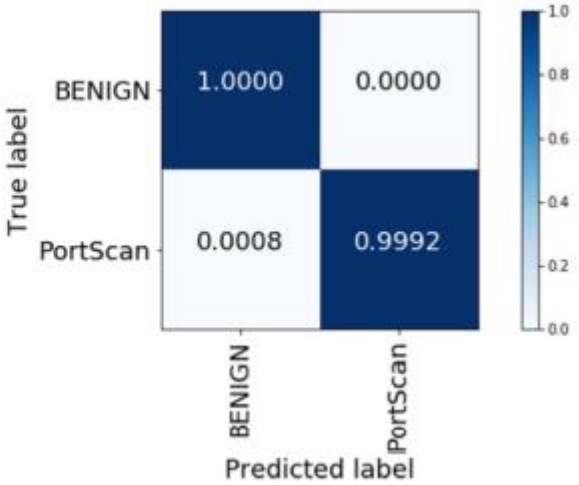


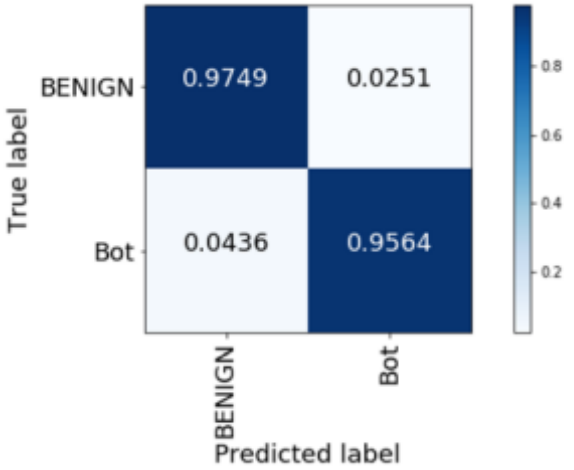


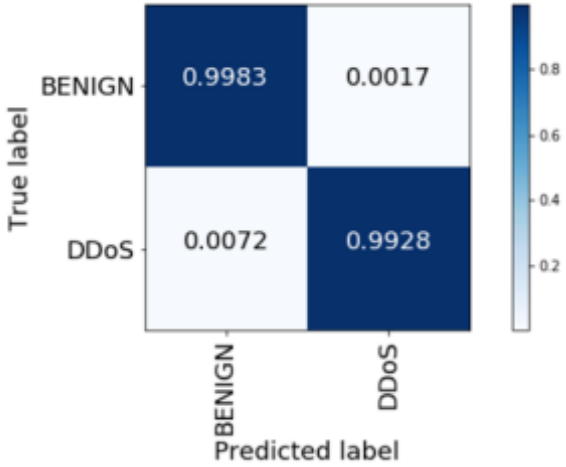
- Benign v.s. DoS, web attack, brute-force attack
- Outliers
  - Heartbleed: only 11 samples
  - SQL injection & XSS: hard to classify, but the are “not benign”

Performance results









- Show high accuracy
- However, the results are “separated”
  - The authors made a classifiers for each attack type
- This means, it seems that hard to distinguish attack types in DDoS
  - And also as combines of web attacks, DoS, and DDoS

Conclusion

- Authors proposed TempoCode-IoT
  - A temporal codebook-based encoding of flow features
  - A novel feature transformation of network flow features based on capturing the key patterns of benign traffic in a learnt temporal codebook

- The experimental evaluations on recent realistic datasets (CICIDS2017 and NBaIoT) proved the effectiveness of TempoCode-IoT representations