Effective Monitoring, Detection and Visualization of Malicious CAN traffic

Sangyup Lee, Shahroz Tariq, Homin Yoon and Simon Woo

SUNY Korea Computer Science Department

정보보호 R&D 데이터 챌린지
차량이상징후 탐지 트랙
2-17-12-08
KISA

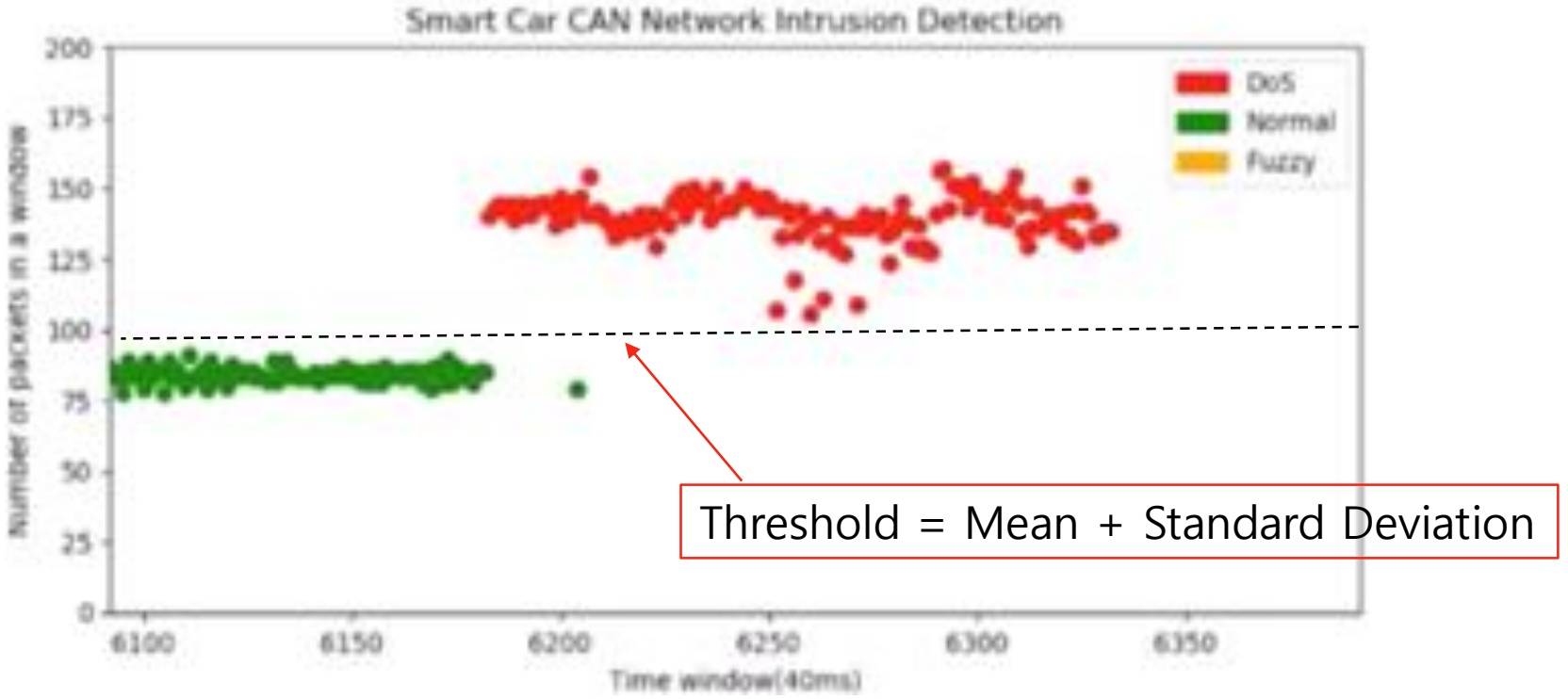
SUNY Korea
The State University of New York

CONTENTS

1. 트래픽 분석

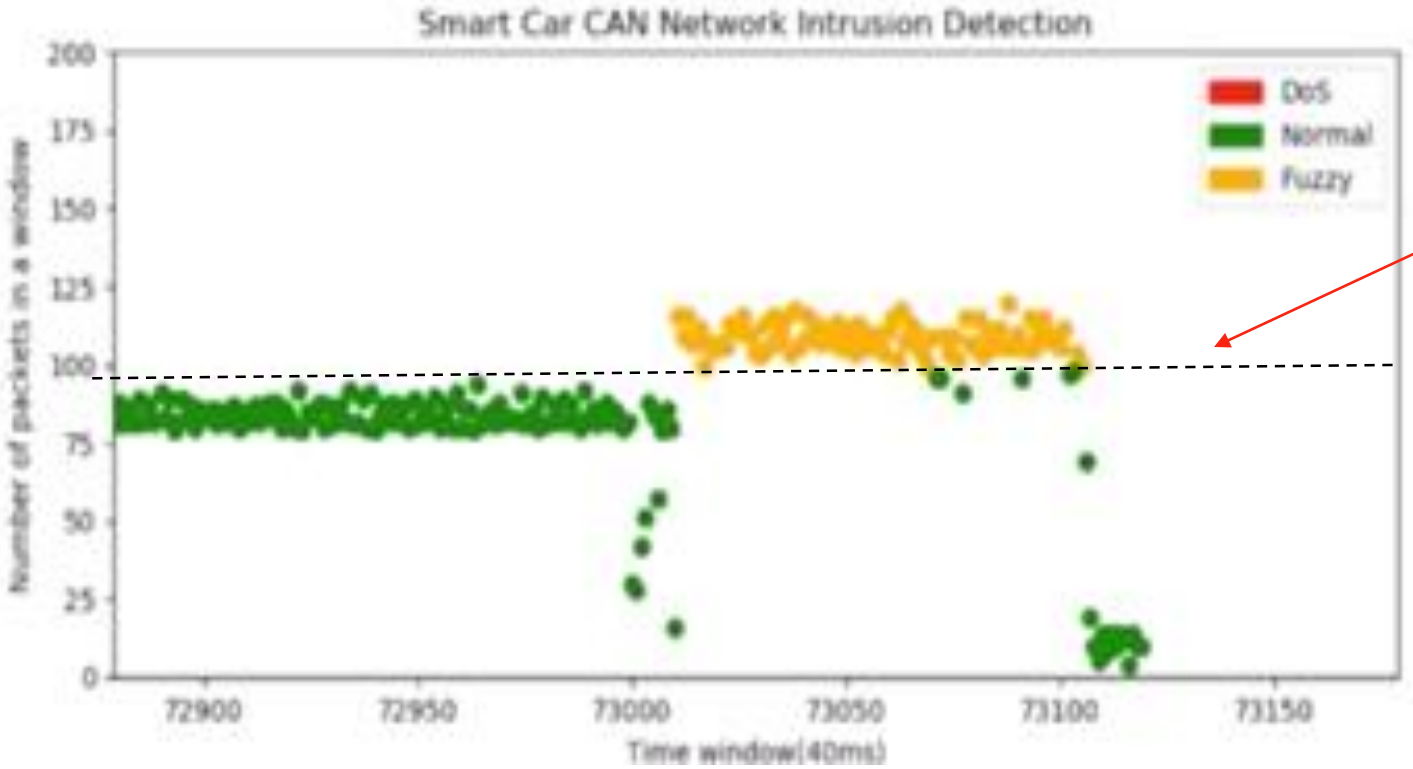
- DoS Attack 분석
- Fuzzy Attack 분석
- Replay Attack 분석
- 군집화를 통한 분석

DoS Attack 분석



- 정상 패킷 수 < 공격 패킷 수
- ID : 우선순위 높음
- 데이터 부분 : 00 00 ... 00 00

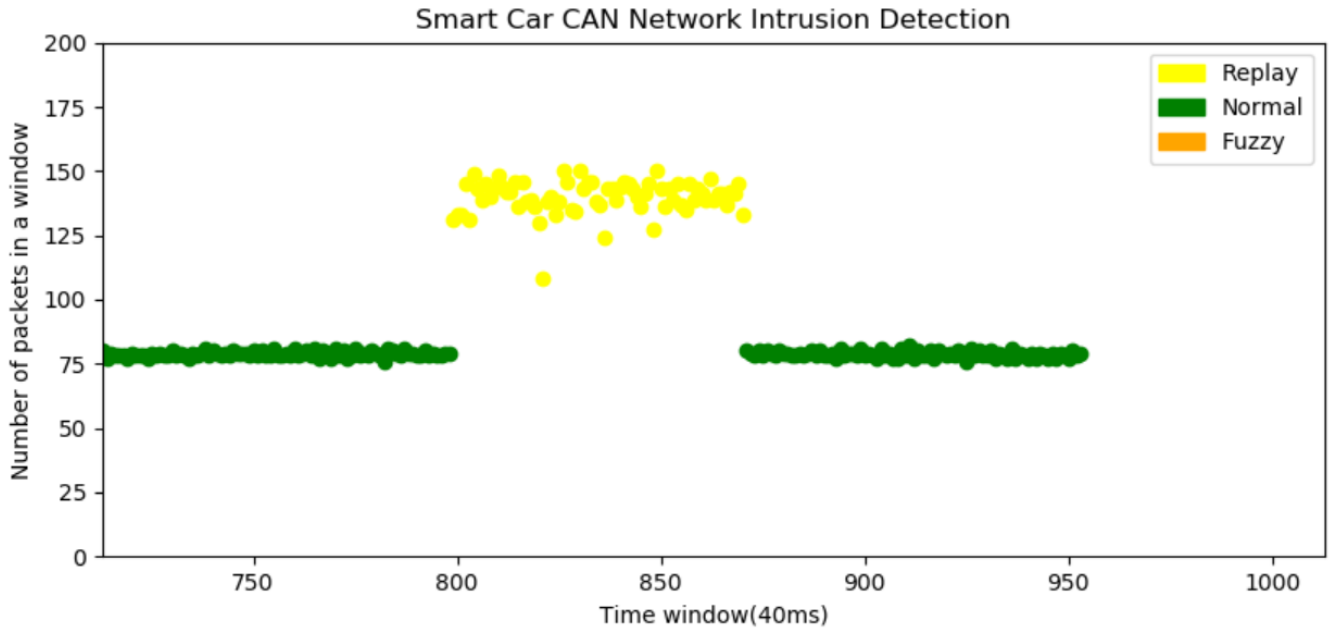
Fuzzy Attack 분석



Threshold = Mean + Standard Deviation

- 정상 패킷 수 < 공격 패킷 수
- ID : 랜덤
- 데이터 부분 : 랜덤

Replay Attack 분석



- 정상 패킷 수 < 공격 패킷 수
- ID : Repeated IDs
- 데이터 부분 : Same Benign Payload Data

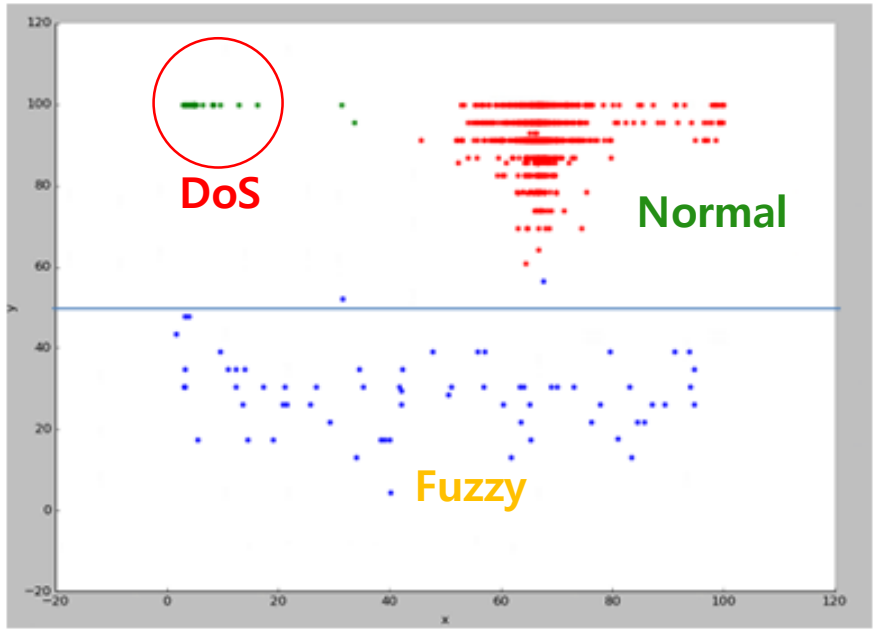
군집화 데이터 분석

SUNY Korea
The State University of New York

K-평균 알고리즘을 통한 군집화
(Unsupervised K-means Clustering)

데이터 부분 분석

Y : 데이터 유사성

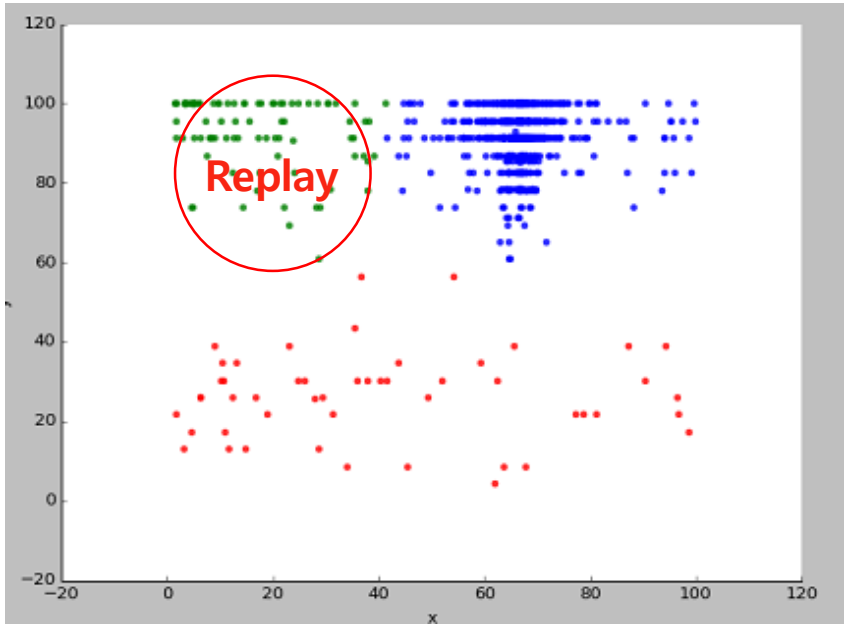


X : 패킷 간 시간 간격

예선 데이터 셋

데이터 부분 분석

Y : 데이터 유사성



X : 패킷 간 시간 간격

본선 데이터 셋

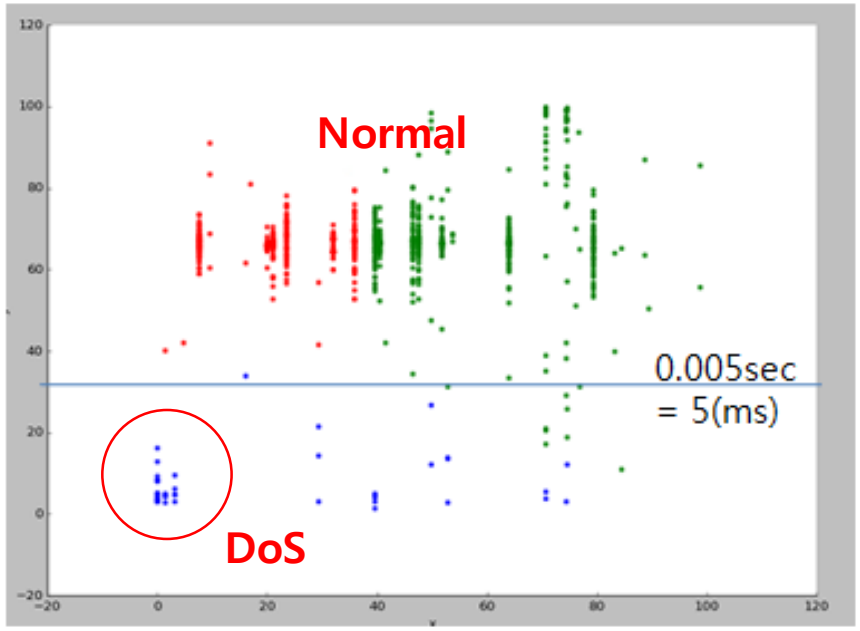
군집화 데이터 분석

K-평균 알고리즘을 통한 군집화 (Unsupervised K-means Clustering)

SUNY Korea
The State University of New York

패킷 간 시간 간격

Y : 패킷 간 시간 간격

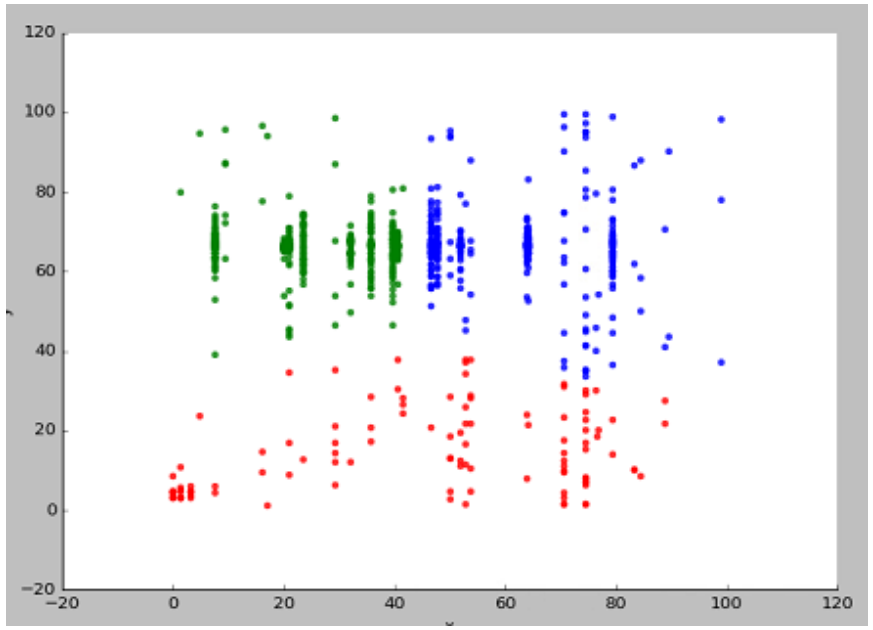


X : ID

예선 데이터 셋

패킷 간 시간 간격

Y : 패킷 간 시간 간격

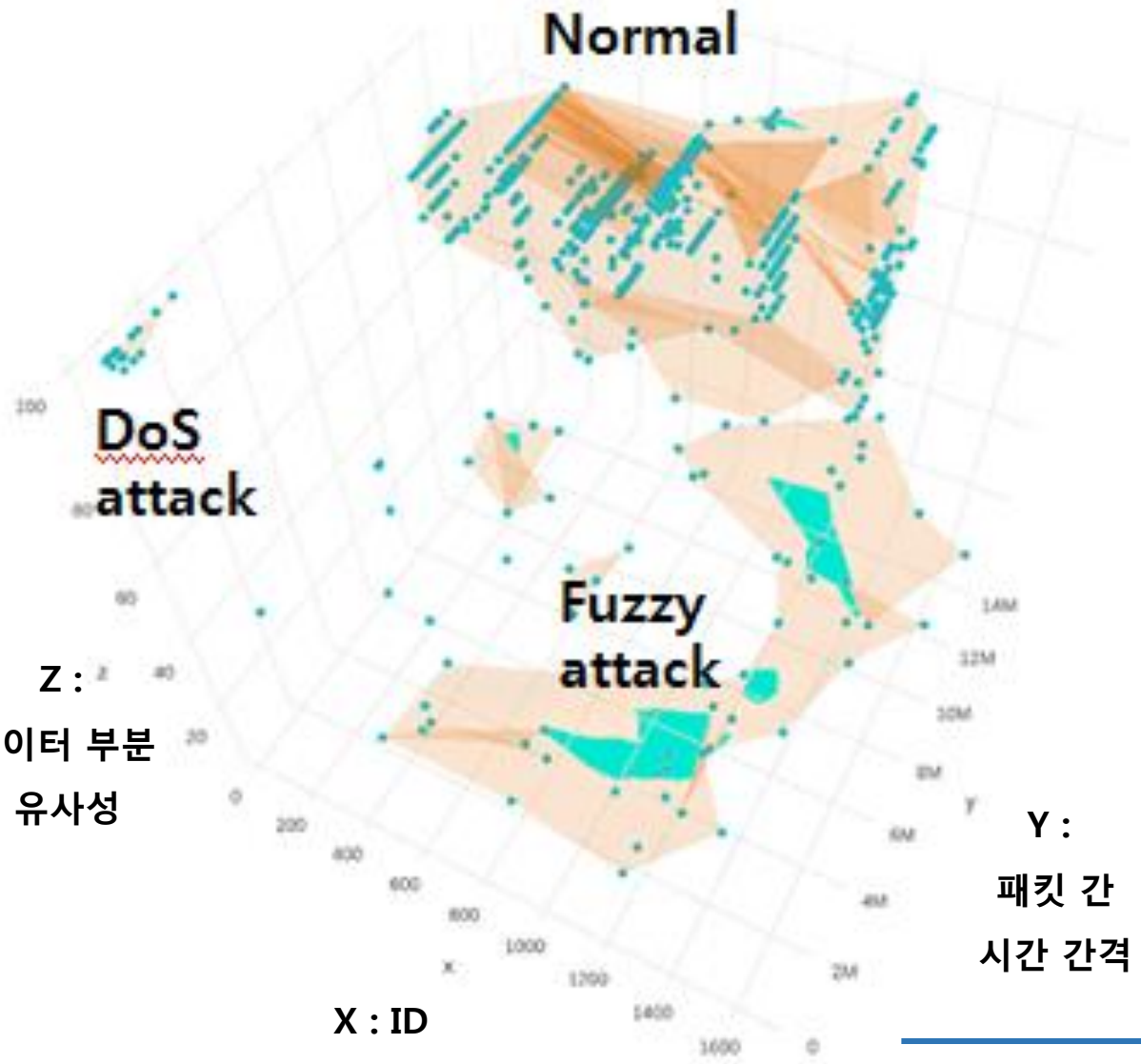


X : ID

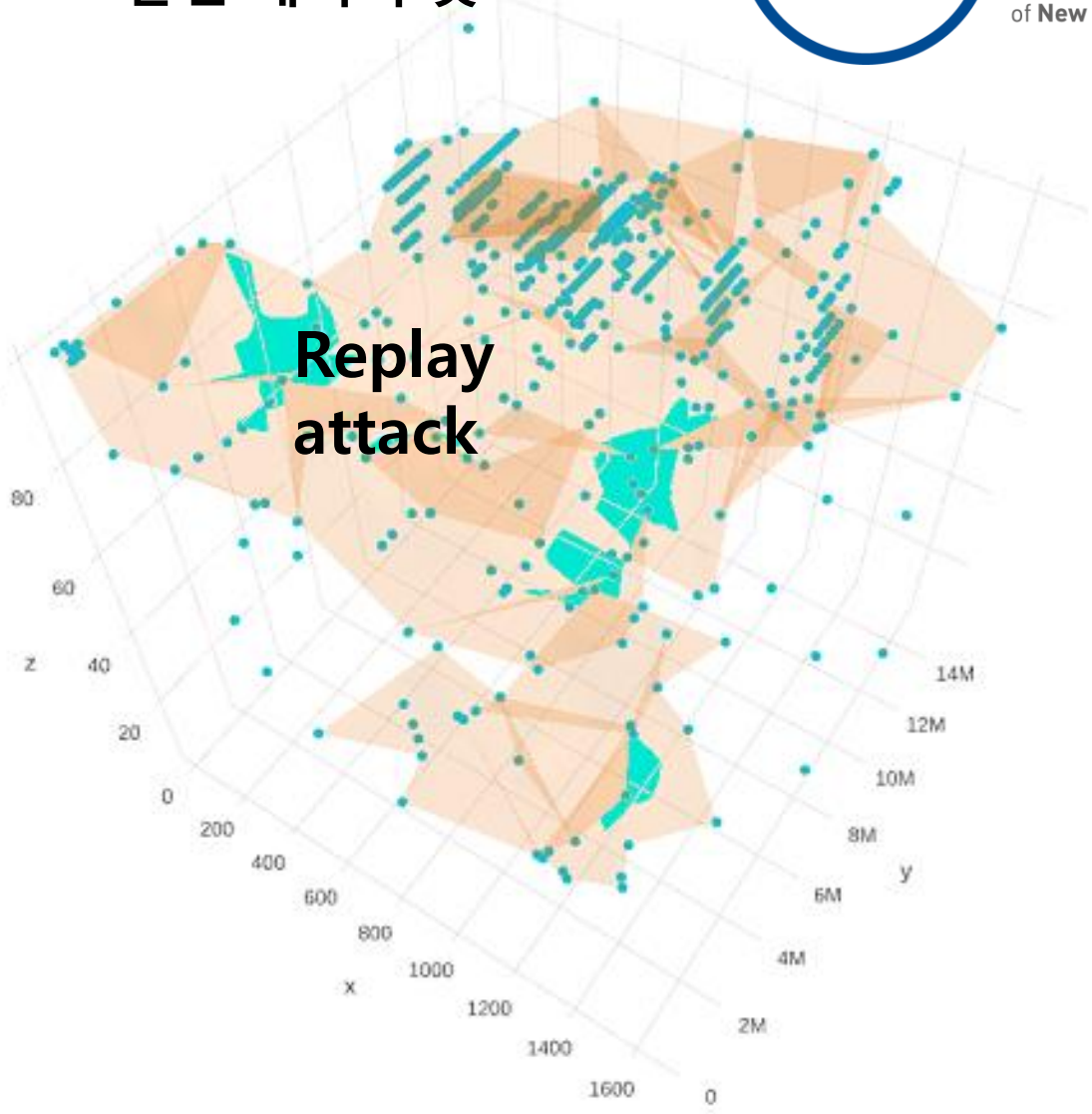
본선 데이터 셋

군집화 데이터 분석

예선 데이터 셋

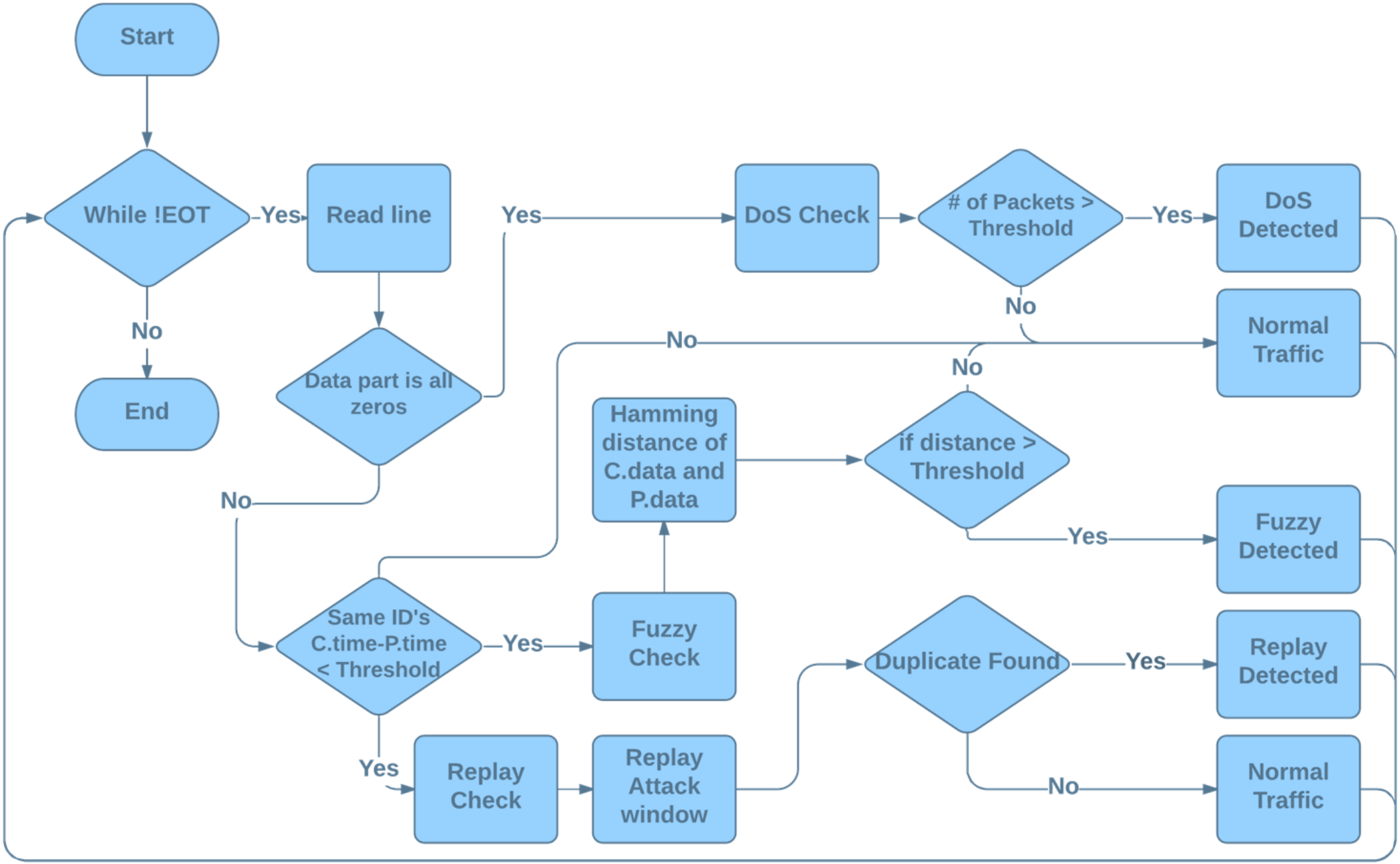


본선 데이터 셋



2. 알고리즘

- 알고리즘 구조
- Pseudo 코드


Start
While !EOT
Yes
Read line
No
End
Data part is all zeros
Yes
No
DoS Check
# of Packets > Threshold
Yes
DoS Detected
No
Normal Traffic
No
Hamming distance of C.data and P.data
if distance > Threshold
Yes
Fuzzy Detected
Same ID's C.time-P.time < Threshold
Yes
Fuzzy Check
Yes
Replay Check
Replay Attack window
Duplicate Found
Yes
Replay Detected
No
Normal Traffic

Pseudo 코드

```
Function FindDoSFuzzy (File):
Foreach Row in File do
        If Row[data] equals '00 00 00 00 00 00 00 00' then
                If Row[id] not in Sthen
                        S[Row[id]] ← Row[Timestamp]
                        D.add(Row[id])
                        C[Row[id]] ← 0
                End if
                E[Row[id]] ← Row[Timestamp]
                C[Row[id]] ← C[Row[id]] + 1
                P[Row[id]] ← Row[0]
    Else
                foreach d in D do
                        if Row[timestamp] - E[d] >DoS_Wait_Threshold
                          And C[d]>DoS_Packet_Thresholdthen
                                Print interval & Attack type as DoS
                                Delete S[d],L[d],C[d],D[d]
                        End if
                End for
```

```
iflength(L[Row[id]]) <List_Threshold
   And Row[data] not in L[Row[id]] then
        L[Row[id]].add(Row[Data]])
End if
if Row[id] not in P[id] then
        P[Row[id]] ←  Row[Data]
End if
If Row[Timestamp] - P[Row[id] >Fuzzy_Thresholdthen
        Foreach l in L[Row[id] do
                Dist ← Dist + Levenshtein_hamming(l,Row[data])
        End for
        IfDist<Fuzzy_Hamming_Thresholdthen
                Print interval & Attack type as Fuzzy
        Else
                L.pop()
                L.add(Row[data])
        End if
    End if
End if
End for
```

3. 실험 및 결론

- 실험
- 결론

Analysis & Experiment
(Dataset 분석 & 실험)

SUNY Korea
The State University of New York

분석/실험 환경: Window, Python3.6.1, Matplotlib 2.1.0

분석/실험 방법:

분석/실험 1. 특정 시간동안 들어오는 주입되는 패킷 수 측정 및 분석 (Data Characteristics)

분석/실험 2. 데이터 (payload) 부분의 유사성 측정 및 분석

분석/실험 3. Replay 데이터 유형 분석

분석/실험 1 결과 - 특정 시간동안 들어오는 주입되는 패킷 수 측정 및 분석

SUNY Korea
The State University of New York

Window Size = 1 sec

Data section	Whole Dataset	During DoS	During Fuzzy	During Replay
Number of windows	3952	13	13	4
Median of # of packets	2089	3031	2791	3212
Mean of # of packets	1909	2959	2604	2610
Standard deviation of # of packets	453	413	685	1139

- DoS, Fuzzy 공격때 한 윈도우당 들어오는 패킷 수의 Median과 Mean값이 증가하는 것을 볼 수 있다.
- Replay 도 DoS 와 비슷하지만 ID 가 Repeat 되는 것을 발견을 할수 있음

분석/실험 2 결과 - 데이터 부분의 유사성 측정 및 분석

SUNY Korea
The State University of New York

정상 패킷

데이터 부분	데이터 빈도	ID 0329 내 데이터 차지 비율(%)
40 b3 80 8c 11 2f 00 10	58467	16
85 b3 80 8c 11 2f 00 10	58377	16
0f b3 80 8c 11 2f 00 10	58314	16

비정상 패킷(Fuzzy)

데이터 부분	데이터 빈도	ID 0329 내 데이터 차지 비율(%)
d7 b3 80 8c 11 35 01 10	1	0.0002
94 5c f1 01 66 17 86 f7	1	0.0002
85 af 80 8c 11 36 00 10	1	0.0002

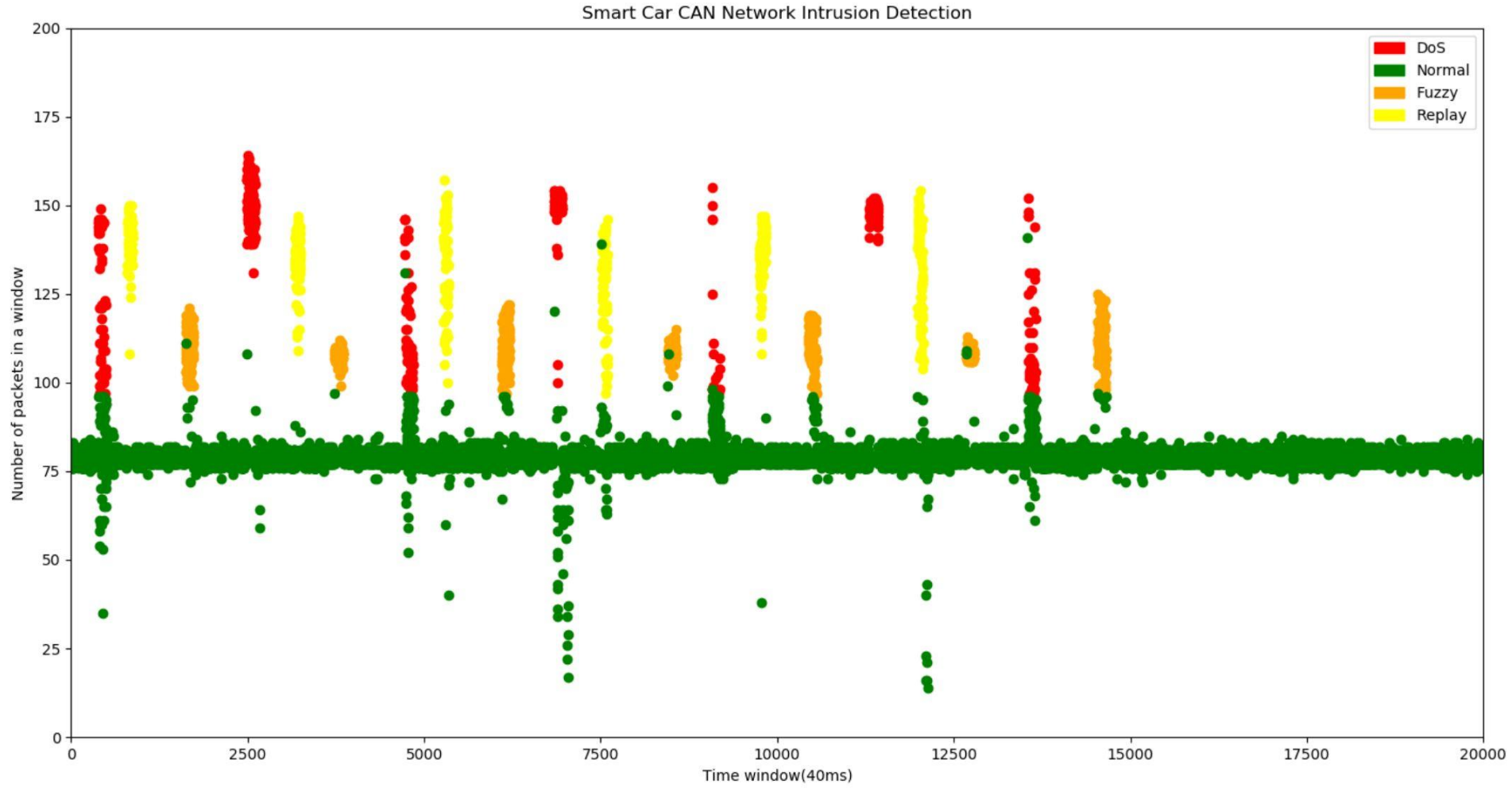
분석/실험 3 결과 - Replay 데이터 유형 분석

SUNY Korea
The State University of New York

Timestamp: 1479225979.224559	ID: 04f0	000	DLC: 8	00 00 00 80 00 64 03 14
Timestamp: 1479225979.224900	ID: 0130	000	DLC: 8	f7 7f 00 ff 10 80 02 c9
Timestamp: 1479225979.225142	ID: 0130	000	DLC: 8	da 7f 00 ff 01 80 02 d7
Timestamp: 1479225979.225368	ID: 0131	000	DLC: 8	12 80 00 00 86 7f 02 de
Timestamp: 1479225979.225608	ID: 0131	000	DLC: 8	d7 7f 00 00 74 7f 02 17
Timestamp: 1479225979.225878	ID: 0140	000	DLC: 8	00 00 00 00 02 12 22 98
Timestamp: 1479225979.226089	ID: 0140	000	DLC: 8	00 00 00 00 04 0f 22 12
Timestamp: 1479225979.226331	ID: 018f	000	DLC: 8	fe 21 00 00 00 58 00 00

Final Detection Result (오늘의 결과)

SUNY Korea
The State University of New York



결론 (Conclusion)

◆ 우리가 제안한 CAN data traffic 알고리즘은 실시간으로 들어오는 normal, DoS, 그리고 Fuzzy, Replay 공격들을 효과적으로 분류할 수 있었다.

◆ 제안된 알고리즘을 통해, ID, 데이터, 트래픽 양을 특정 윈도우 상에서 실시간 측정하여 의심가는 트래픽 패턴을 찾을 수 있다.

◆ 구현한 시각화 프로그램은 분석된 결과를 실시간으로 보여줄 뿐만 아니라 새로운 공격 패턴을 보여주고 탐지하는 데도 도움이 될 것이다.

◆향후 이 알고리즘을 향상시키고 기계학습 (SVM, Random Forest, LSTM)을 응용한 방법을 동시에 적용하여 unknown 공격 패턴을 탐지하는 연구를 수행할 예정이다.


Q & A