Chapter 8
EARLY - a tool for real-time security attack detection

Tanwir Ahmad, Dragos Truscan - Åbo Akademi University, Finland
Jüri Vain - Tallinn University of Technology, Estonia

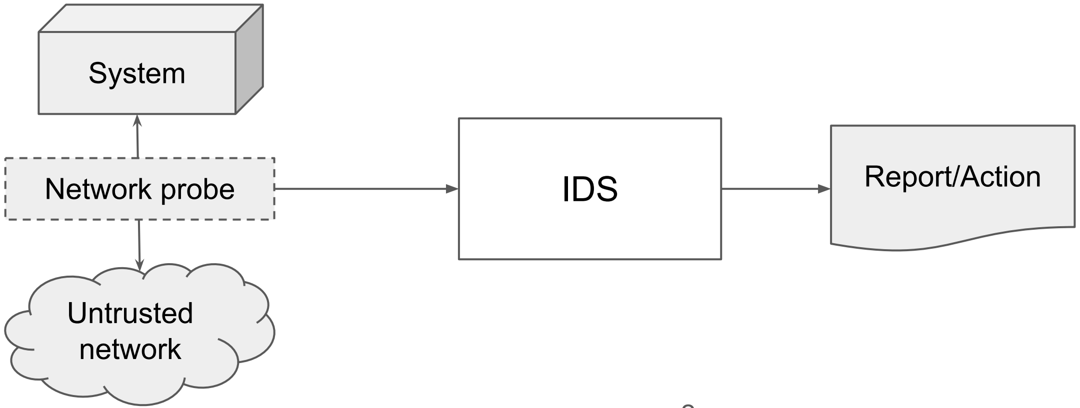
Åbo Akademi University

VeriDevOps

Network Intrusion Detection System (IDS)

Identify unauthorized and **malicious behavior** by **observing** the **network** traffic.

Allow network **administrators** take appropriate **preventive measures** to **secure** the network **infrastructure** and the associated **nodes.**



Åbo Akademi University

VeriDevOps

Types of network IDSs

- Anomaly based
  - Differentiate between normal and anomalous network traffic
  - Allow to discover novel attacks

- Signature based
  - Compare network traffic against signatures of known attacks
  - Allows the administrator to deploy specific countermeasures depending of the attack type
  - One challenge is in extracting and defining the signature of a known attack that can detect variations of the attack

Åbo Akademi University

VeriDevOps

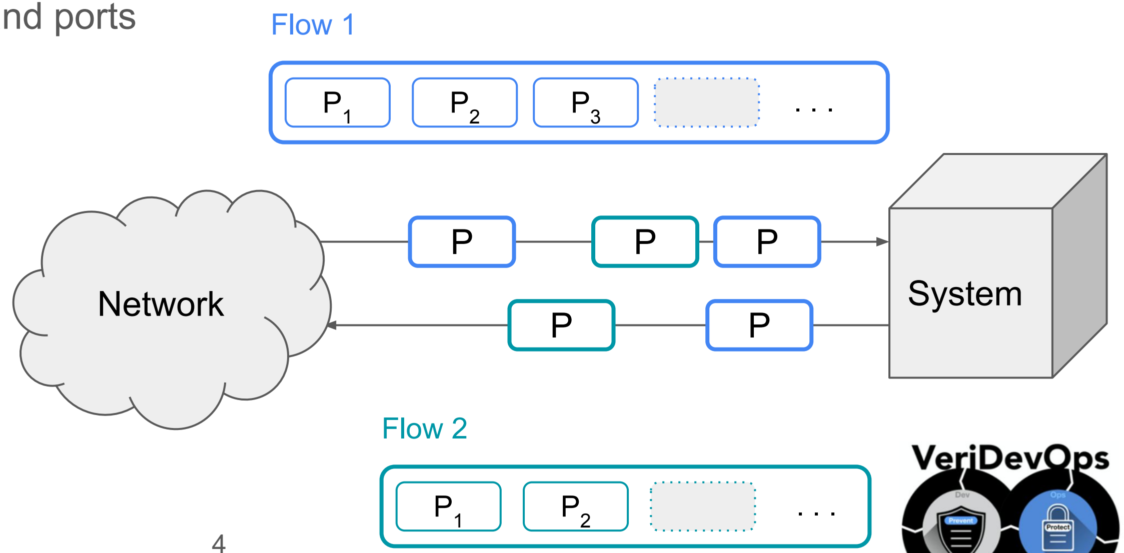
Flow-based Network IDSs

Most of the **state-of-the-art IDSs** utilize network flows for attack detection.

Flow is a **sequence** of packets between 2 endpoints

Packets are **grouped** into **flows** based on the:

- Source and destination addresses and ports
- Protocol type
- Time interval



Åbo Akademi University

VeriDevOps

Network IDS

- Extract **flow-based statistical features** by analyzing **all** the packets in a flow such as:
  - total bytes,
  - packets count,
  - IP addresses and ports numbers.

- Learn to identify attacks using those statistical features.

Motivation

- **Problem**
  - Current IDSs **detect** attacks by **inspecting** the **complete information** about the attack.
  - **After** the **attack** has been **executed** on the system under attack.

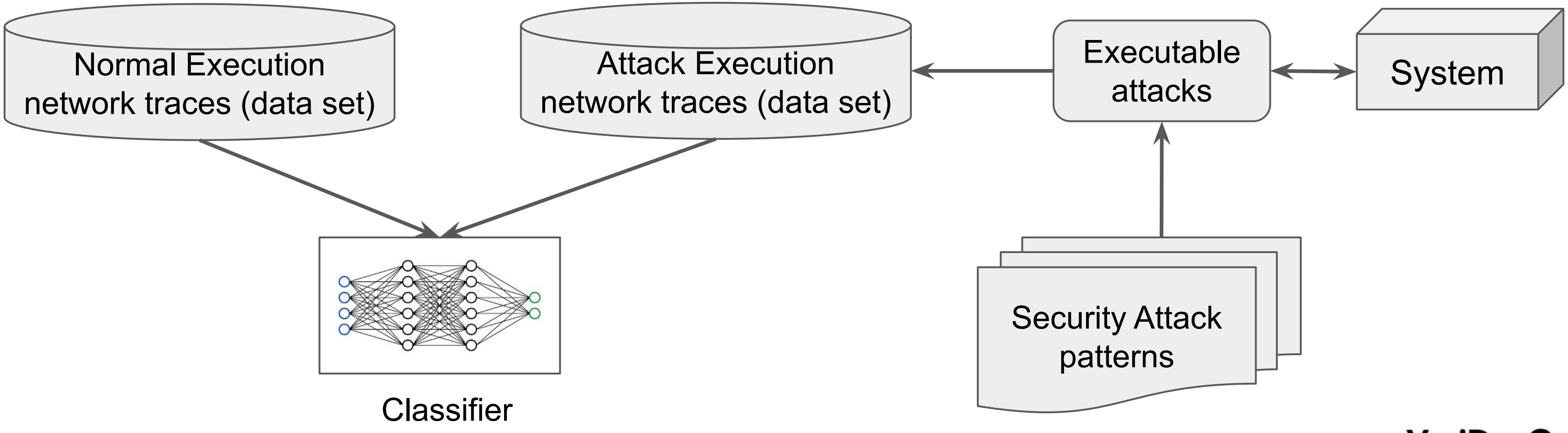
- **Research Objective**
  - **Identify** network attacks as **early as possible** by monitoring the network traffic in real-time.
  - Allow to deploy countermeasures **before the attack completes**

Åbo Akademi University

VeriDevOps

Overview of the approach

Stage 1: training

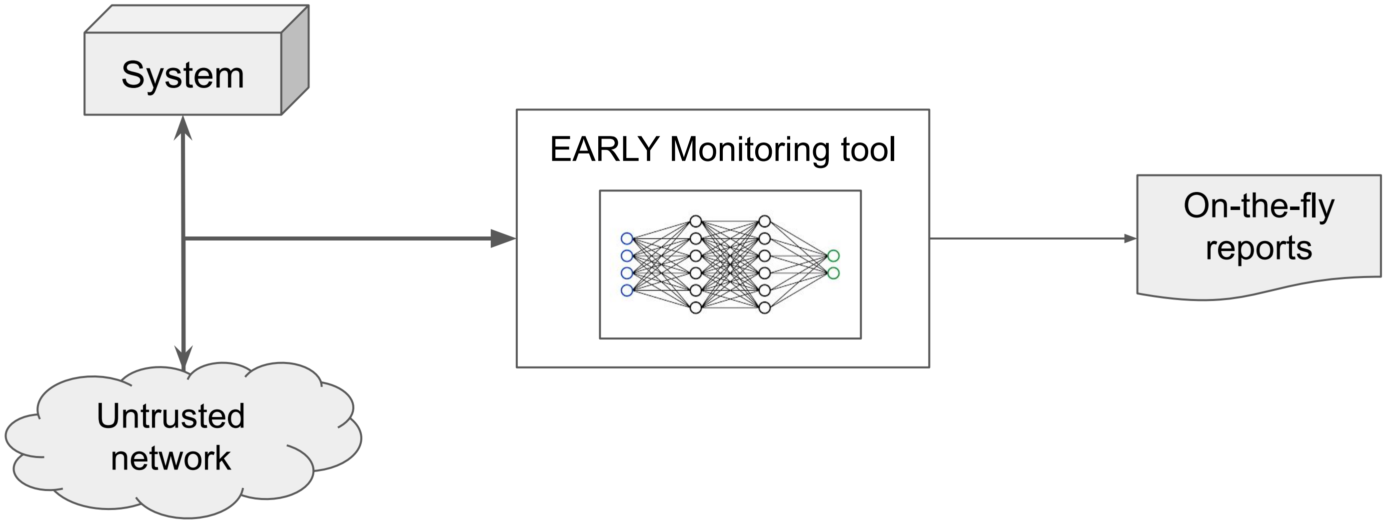


Åbo Akademi
University

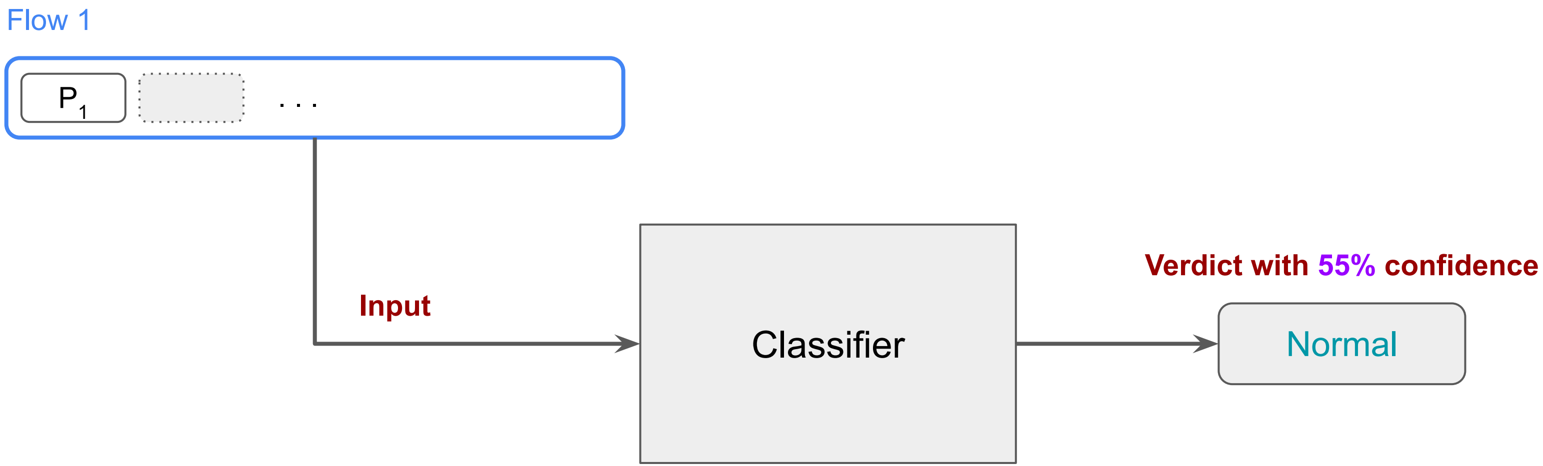
VeriDevOps

Overview of the approach

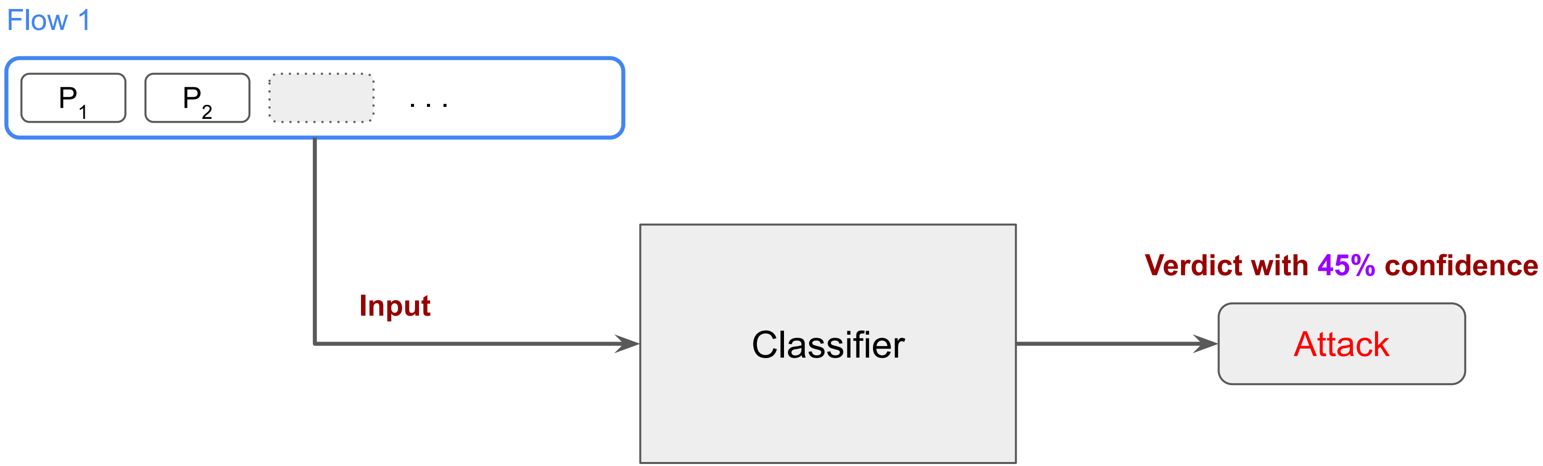
Stage 2: monitoring



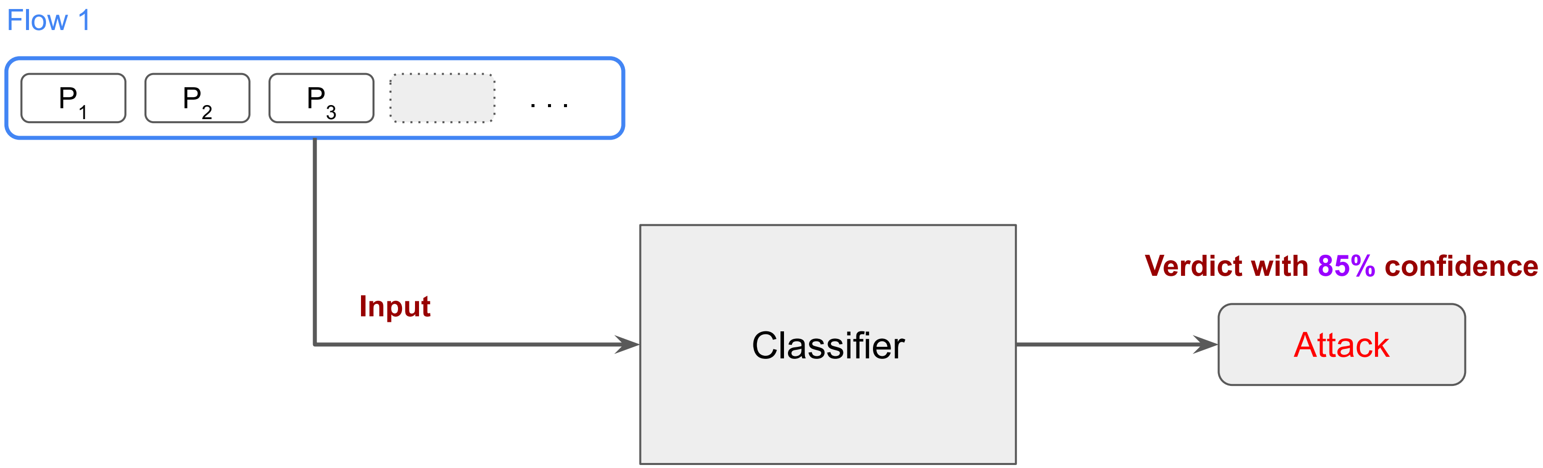
Monitoring Network for Attacks



Monitoring Network for Attacks

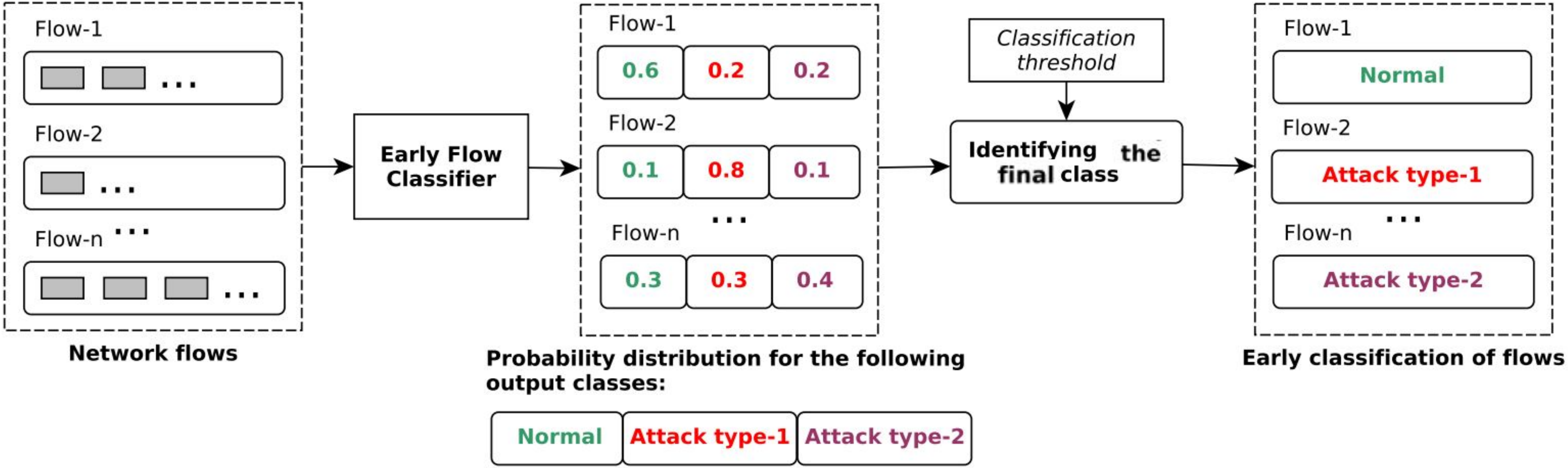


Monitoring Network



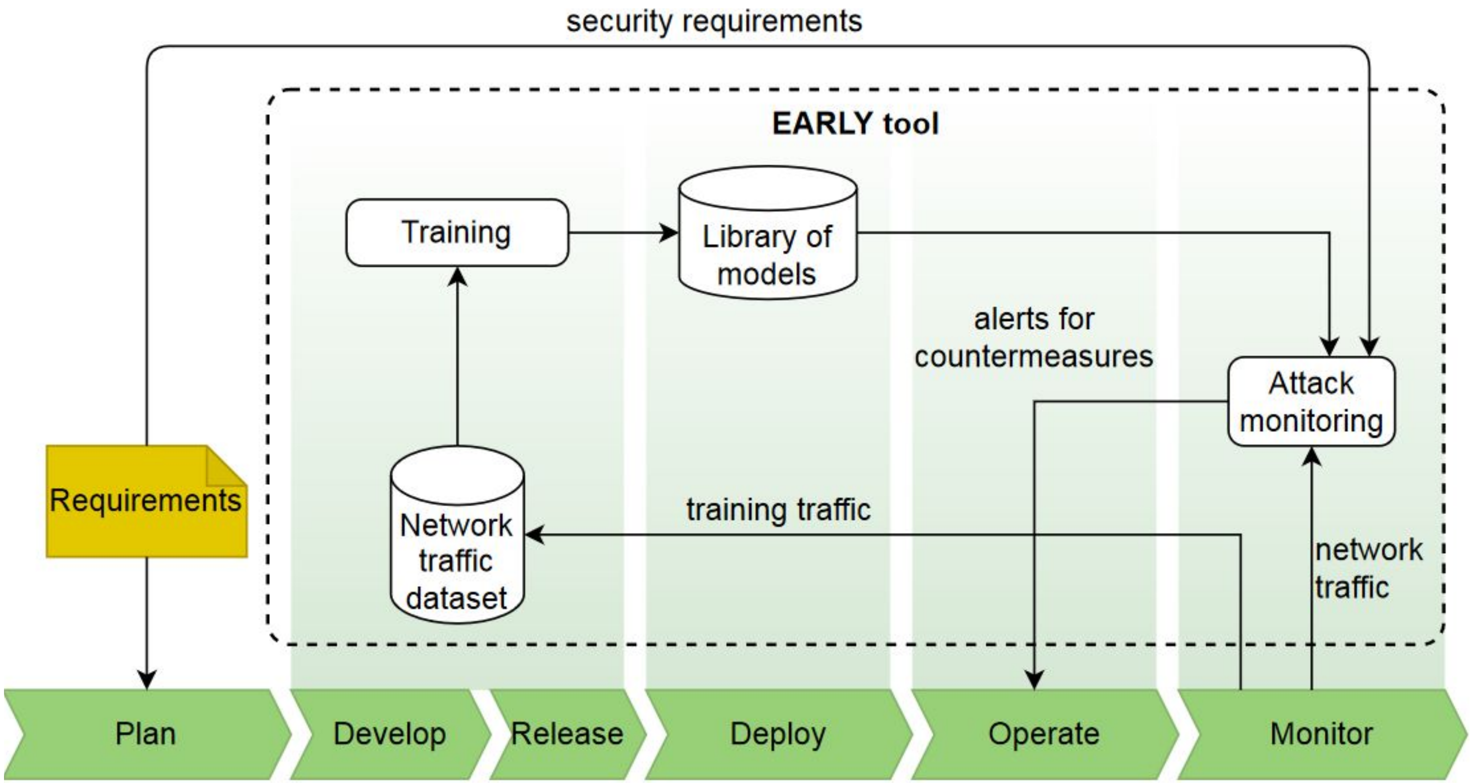
Monitoring Network for Attacks

Flow ID	Source IP	Destination IP	Length	Prediction	Confidence	Remarks
Flow 3	172.16.0.1	192.168.10.50	2	XSS	100.0	ALERT
Flow 2	172.16.0.1	192.168.10.50	12	Brute Force	100.0	ALERT
Flow 1	172.16.0.1	192.168.10.50	14	XSS	99.0	ALERT
Flow 0	192.168.10.15	131.253.61.98	5	Normal	100.0	



Åbo Akademi University

Integration with DevOps Environments



Åbo Akademi University

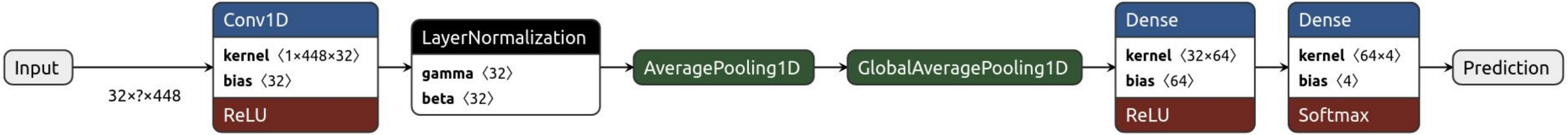
VeriDevOps

Evaluation

- Neural network architectures
  - 1- Dimensional Convolution Neural Network
  - Recurrent Neural Network
- Datasets
  - CICIDS-2017
  - MQTT-IoT-IDS-2020

Evaluation

- 1- Dimensional Convolution Neural Network (EARLY$_{CNN}$)



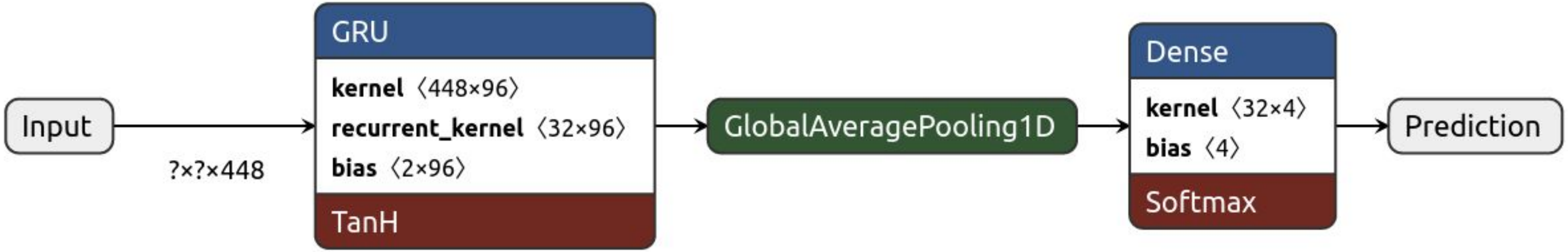
- Parameters: 16,804
- Epoch: 50
- Training time: 10 minutes
- Machine: i9 with RTX 3090

Åbo Akademi University

VeriDevOps

Evaluation

- Recurrent Neural Network ($EARLY_{RNN}$)



- Parameters: 46,404
- Epoch: 50
- Training time: 30 minutes
- Machine: i9 with RTX 3090

Åbo Akademi University

VeriDevOps

Evaluation

Dataset: CIC-IDS2017

Class	Number of Flows	Avg. Flow Length
Normal	27,129	124.39
Brute force	1,507	18.43
XSS	652	11.48
SQL Injection	21	5.71

70% of the data for training and 30% for testing.

10-fold cross-validation to fine-tune the hyper-parameter values and model selection.

Åbo Akademi University

VeriDevOps

Evaluation

Dataset MQTT-IDS-2020

Class	Number of Flows	Avg. Flow Length
Normal	363,495	5.81
Brute force	2,000,211	4.99
Aggressive scan	20,025	2.03
UDP scan	10	1.10
Sparta SSH	1,013,380	19.45

70% of the data for training and 30% for testing.

10-fold cross-validation to fine-tune the hyper-parameter values and model selection.

Åbo Akademi University

VeriDevOps

Evaluation metrics

Precision - What proportion of positive identifications was actually correct?

Recall - What proportion of actual positives was identified correctly?

False positive rate (FPR) - proportion of negative observations wrongly predicted as positive over the total number of negative observations.

Earliness - after how many packets in a flow we can classify an attack

Classification Performance (CICIDS-2017)

Class	Precision		Recall		FPR	
	CNN	RNN	CNN	RNN	CNN	RNN
Normal	0.996	0.996	0.944	0.995	0.054	0.052
Brute force	0.720	0.905	0.828	**0.916**	0.051	0.003
XSS	0.754	0.823	**0.911**	**0.916**	0.008	0.004
SQL Injection	0.343	0.403	0.528	0.733	0.003	0.001

Balanced Accuracy: `0.803 (CNN) < 0.890 (RNN)`

Åbo Akademi University

VeriDevOps

Classification Performance (MQTT-IDS-2020)

	Precision		**Recall**		**FPR**	
Class	**CNN**	**RNN**	**CNN**	**RNN**	**CNN**	**RNN**
Normal	0.707	<u>0.827</u>	0.584	<u>0.758</u>	0.095	<u>0.053</u>
Brute force	0.979	<u>0.995</u>	**0.997**	**<u>0.999</u>**	0.008	<u>0.002</u>
Aggressive scan	0.812	<u>0.938</u>	0.815	<u>0.987</u>	0.055	<u>0.022</u>
UDP scan	0.004	<u>0.092</u>	<u>0.422</u>	0.211	0.038	<u>0.000</u>
Sparta SSH	0.809	<u>0.833</u>	0.778	<u>0.853</u>	0.066	<u>0.058</u>

Balanced Accuracy: `0.719 (CNN) < 0.762 (RNN)`

Åbo Akademi University

VeriDevOps

Earliness Performance

Earliness metric

$$Earliness = \begin{cases} \dfrac{T - t}{T - 1} & \text{if } T > 1 \\ 1 & \text{if } T = 1 \end{cases}$$

T = total number of packets in a given flow

t = minimum number of packets required to correctly predict the class of a given flow

! this metric is only applied to those flows that are correctly classified and t ≤ T.

Åbo Akademi University

VeriDevOps

Earliness Performance (CICIDS-2017)

Class	Earliness		Avg value of *t*		Avg. Flow Length
	CNN	RNN	CNN	RNN	
Normal	0.991	0.994	2.11	1.74	124.39
Brute force	0.936	0.931	2.11	2.20	18.43
XSS	0.917	0.886	1.86	2.19	11.48
SQL Injection	0.509	0.712	3.31	2.31	5.71

Both models show the same earliness performance

Åbo Akademi University

VeriDevOps

Earliness Performance (MQTT-IDS-2020)

Class	Earliness		Avg value of *t*		Avg. Flow Length
	CNN	RNN	CNN	RNN	
Normal	0.708	<u>0.922</u>	2.40	<u>1.03</u>	5.81
Brute force	0.991	<u>0.999</u>	1.03	<u>1.00</u>	4.99
Aggressive scan	0.848	<u>0.974</u>	1.15	<u>1.02</u>	2.03
UDP scan	<u>0.525</u>	0.467	<u>1.04</u>	1.05	1.10
Sparta SSH	0.689	<u>0.778</u>	6.73	<u>5.09</u>	19.45

Åbo Akademi University

VeriDevOps

Prediction time

2 machines: 1 replay and 1 monitoring

- Intel Core i9-10900X CPU, 64 GB of memory, RTX 3090 graphics card, and Ubuntu 20.04
- 1Gb Ethernet connection

Dataset	Duration (sec)	Packets re-transmitted	Packet IAT (ms)	Architecture	Prediction time (ms)
CICIDS2017	29 004	4 074 195	7.11	$EARLY_{CNN}$	0.06
				$EARLY_{RNN}$	0.42
MQTT-IDS-2020	16 614	32 144 887	0.51	$EARLY_{CNN}$	4.18
				$EARLY_{RNN}$	4.30

Åbo Akademi University

VeriDevOps

Conclusion

EARLY detects in real-time while happening with a certain probability

Tool supports two types of classifier architectures, **CNN** and **RNN** for early attack identification

Empirically evaluated our approach on the **CICIDS-2017** and **MQTT-IDS-2020** datasets

CNN smaller models (4x), faster training 3x, faster predictions

RNN more accurate

Achieve a high degree of accuracy by analyzing roughly only **1 to 3 packets**

Thank you!