

Detection of Denial of Service Attacks Using Echo State Networks

Kamila Bekshentayeva

kdagilov@sfu.ca

July 2021

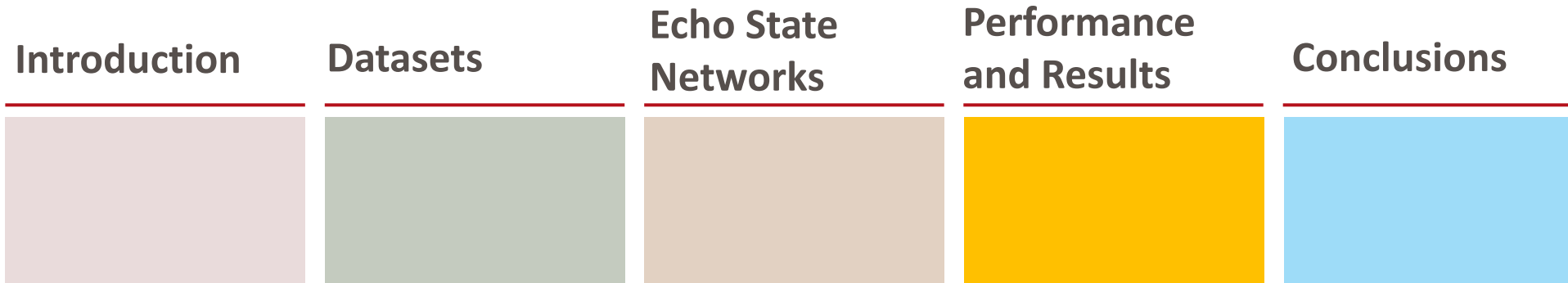
Communication Networks Laboratory

<http://www.ensc.sfu.ca/~ljilja/cnl/>

School of Engineering Science

Simon Fraser University

Roadmap



Roadmap

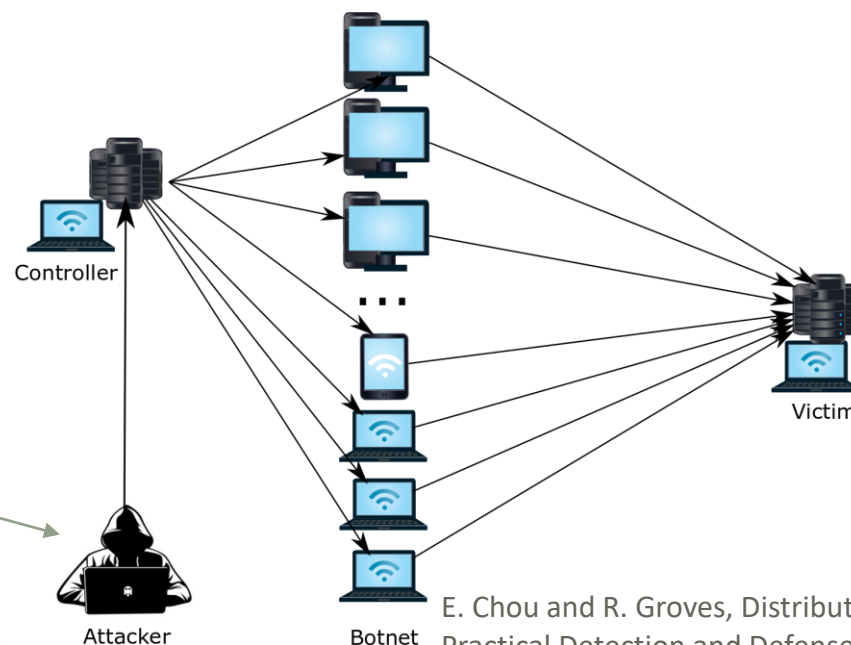
Introduction

- Overview of DoS and DDoS Attacks
- Motivation
- Machine Learning
- Research Contribution

Denial of Service and Distributed Denial of Service (DoS and DDoS): Overview

- **Denial of Service (DoS)** attacks are attempts of an attacker to make services unavailable to legitimate users.
- **Distributed Denial of Service (DDoS)** attacks combine the resources of multiple compromised end systems in a coordinated way to exhaust resources of a target system.

- **ATTACKER:** a cyber criminal, a hacktivist, or a user, who pursues financial gain, prestige, or follows his/her other personal goals.
- He/she **utilizes the best-effort Internet architecture.**

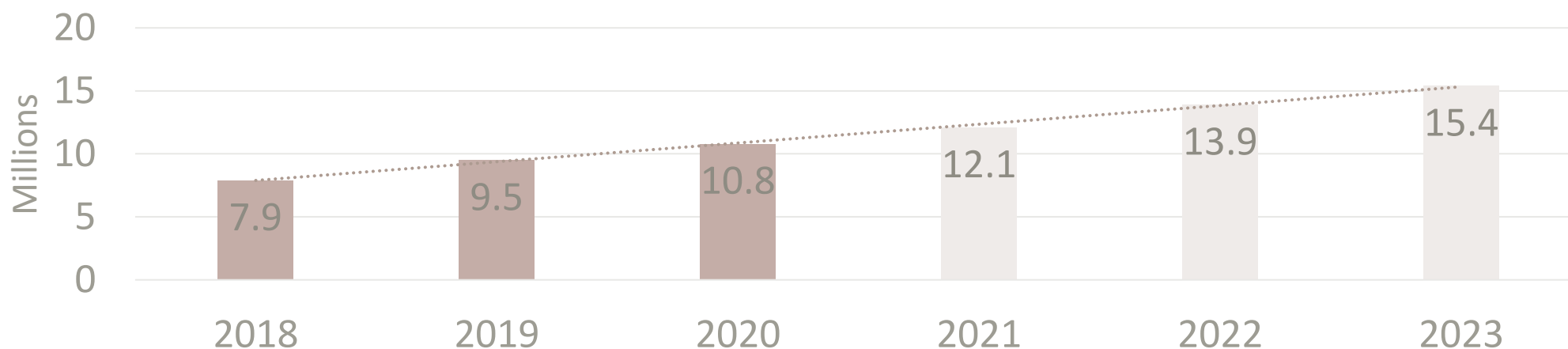


E. Chou and R. Groves, Distributed Denial of Service (DDoS): Practical Detection and Defense. 1st Ed. Sebastopol, CA: O'Reilly Media, 2018.

Motivation: DoS/DDoS are evolving and becoming harder to detect

- The **first documented DDoS** attack occurred in **1999** utilizing 227 bots.
- **Novel attacks strategies:** Internet of Things (Mirai) and artificial intelligence (Github)
- The **largest attack** of all time - DDoS attack of 2.3 Tbps happened in February **2020** that affected **Amazon** cloud services and caused three days of elevated threat.

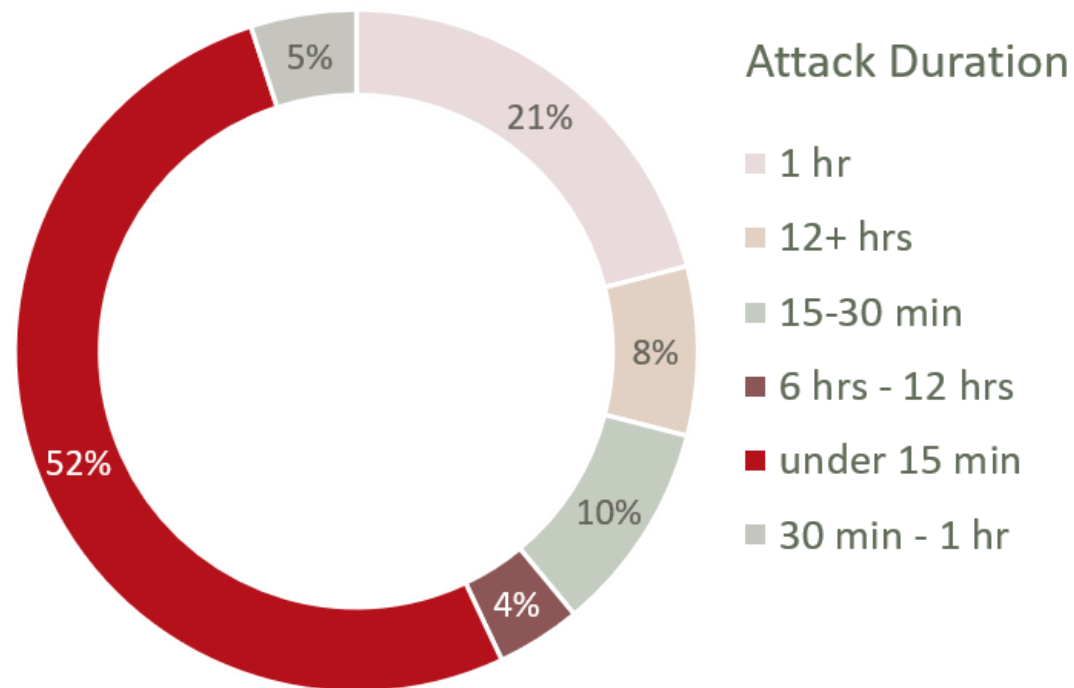
Cisco's analysis of DDoS total attacks: history and predictions.



Cisco Annual Internet Report (2018–2023) White Paper. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.

Motivation: Defense against DoS/DDoS is an important research area

- DoS and DDoS attacks significantly **affect the Internet performance**

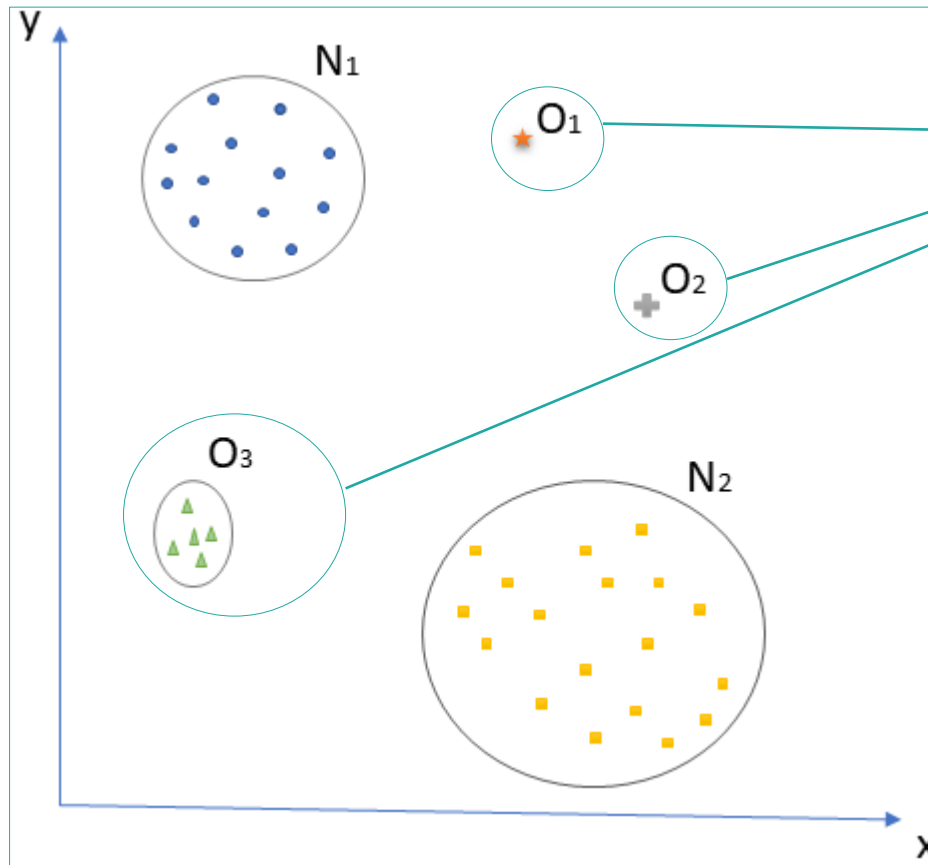


509 hours
longest attack duration

>\$2M USD
cost per DDoS attack on average for enterprises

Cisco Annual Internet Report (2018–2023) White Paper. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.

Anomaly detection refers to identifying the patterns in data that do not conform to expected behavior



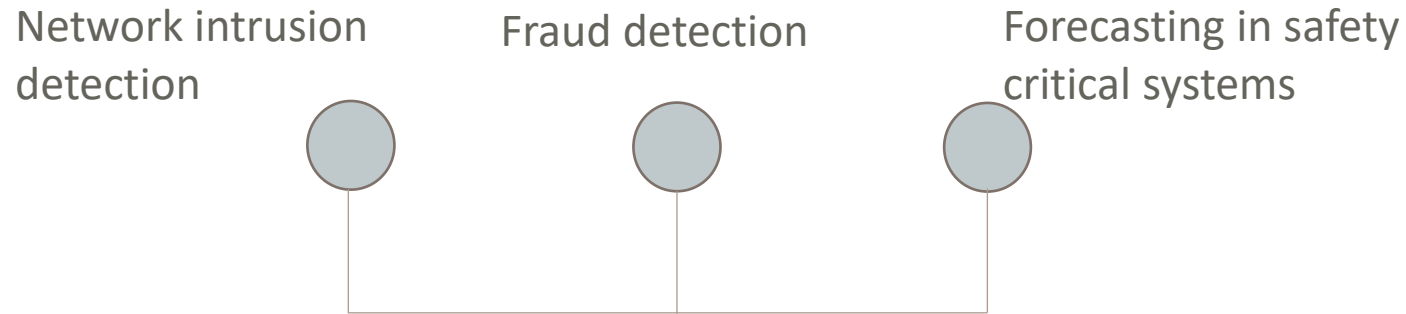
Point and collective anomalies

Challenges of anomaly detection:

- Difficulty of defining normal regions' boundaries
- Varying notions of what anomaly is for different application domains
- Adapting anomalous observations to appear as normal by adversaries
- Lack of labeled data for training

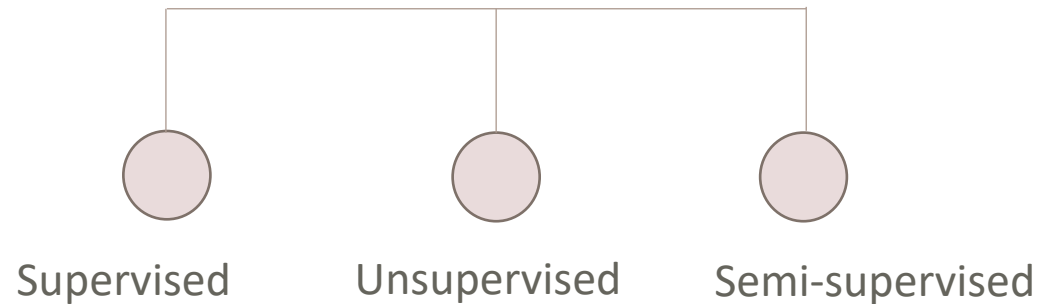
V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," ACM Comput. Surv., vol. 41, no. 3, pp. 15:1–15:58, July 2009.

Anomaly detection has been used in various research areas, such as machine learning, statistics, information theory



Anomaly detection

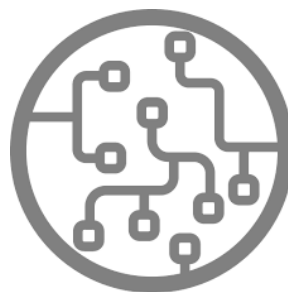
→ Host-based and network-based



V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," ACM Comput. Surv., vol. 41, no. 3, pp. 15:1–15:58, July 2009.

Machine Learning

Involves the design of learning algorithms that optimize their performance as more data are observed to solve a specific task



Various **network anomaly detection systems** employ **machine learning algorithms**: convolutional neural networks, recurrent neural networks (RNNs), deep belief networks, and autoencoders.

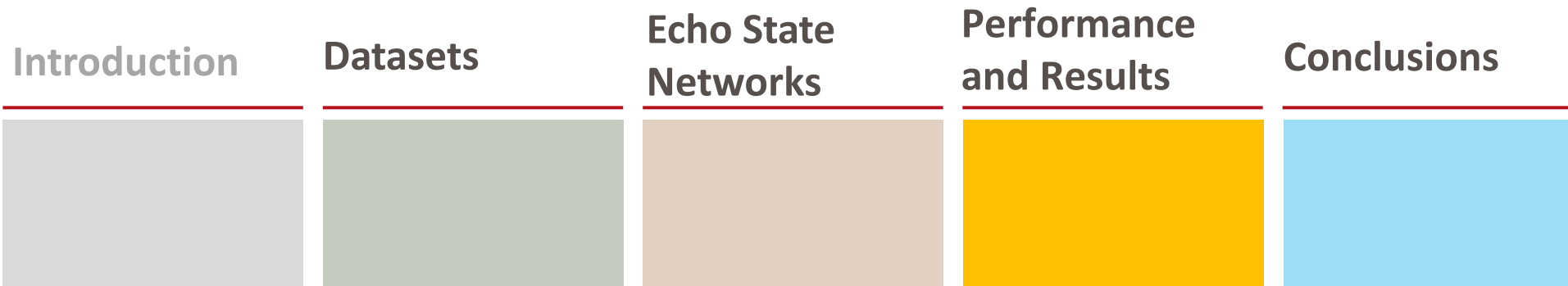
@ SFU Communication Networks Lab:

Support Vector Machines (SVM), Recurrent Neural Networks (LSTM, GRU), Broad Learning System (BLS), deep learning networks, boosting algorithms and decision trees → intrusion detection in network traffic.

Research Contributions

- Echo state networks (ESNs) are used as a **feasible** reservoir computing approach to **identify intrusions in the network. We show they are/they have:**
 - **Not resource intensive** and **simple** to implement (may be used on devices with limited computational/memory resources)
 - Comparable performance with **short training time**
- Investigating how configuration of **reservoir hyperparameters, cross-validation, feature selection** influence the performance of ESN models.
- Models are compared based on **accuracy, F-Score, false alarm rate**, and **training time** to bidirectional long short-term memory (**bi-LSTM**).
- **Employed datasets: CIC-IDS2017, CSE-CIC-IDS2018, CICDDoS2019** (not balanced and **balanced via resampling**) and **Border Gateway Protocol** (Slammer, Nimda, Code Red I worms and recent large DDoS events).

Roadmap



Roadmap

Datasets

- CIC-IDS2017, CSE-CIC-IDS2018, and CIC-DDoS2019 Datasets
- Data Preprocessing: CIC-IDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Border Gateway Protocol Datasets
- Data Preprocessing: BGP Datasets
- Feature Selection

CIC-IDS2017, CSE-CIC-IDS2018, and CIC-DDoS2019 Datasets



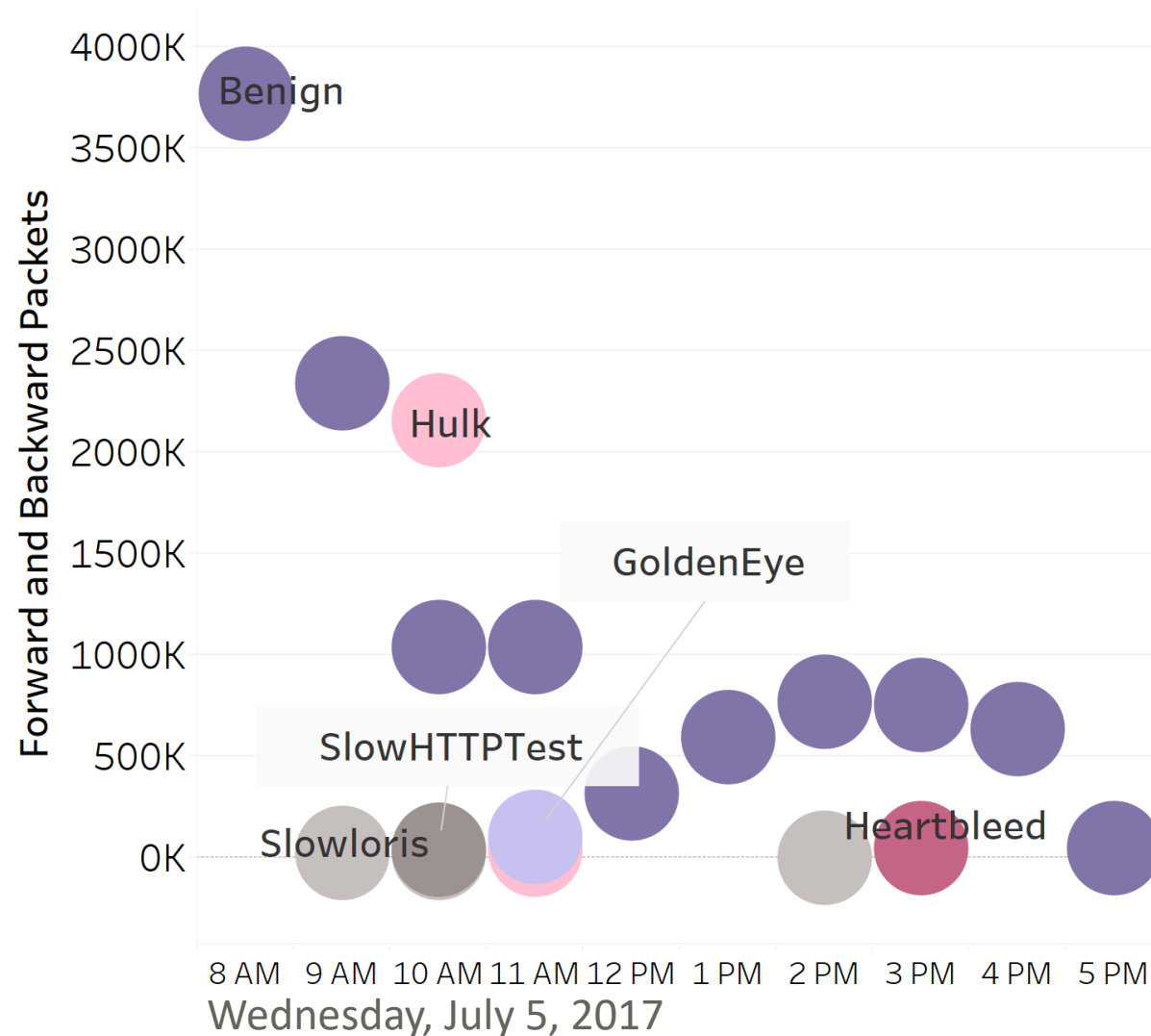
- **Public**
- **Labeled**
- **Diverse features**

- Canadian Institute for Cybersecurity (CIC) → **CIC-IDS2017**, **CSE-CIC-IDS2018** (colab. Communications Security Establishment (CSE)), and **CIC-DDoS2019** datasets with current network traffic trends
- **B-Profile**: background regular behavior of 25 users
 - Protocols: HTTP, HTTPS, FTP, SSH, SMTP, POP3, and IMAP*
- **M-Profile**: infiltration, DoS, web application, and brute force attacks

*HTTP – Hypertext Transfer Protocol; FTP – File Transfer Protocol; SSH – Secure Shell; SMTP – Simple Mail Transfer Protocol; POP3 – Post Office Protocol; IMAP – Internet Mail Access Protocol

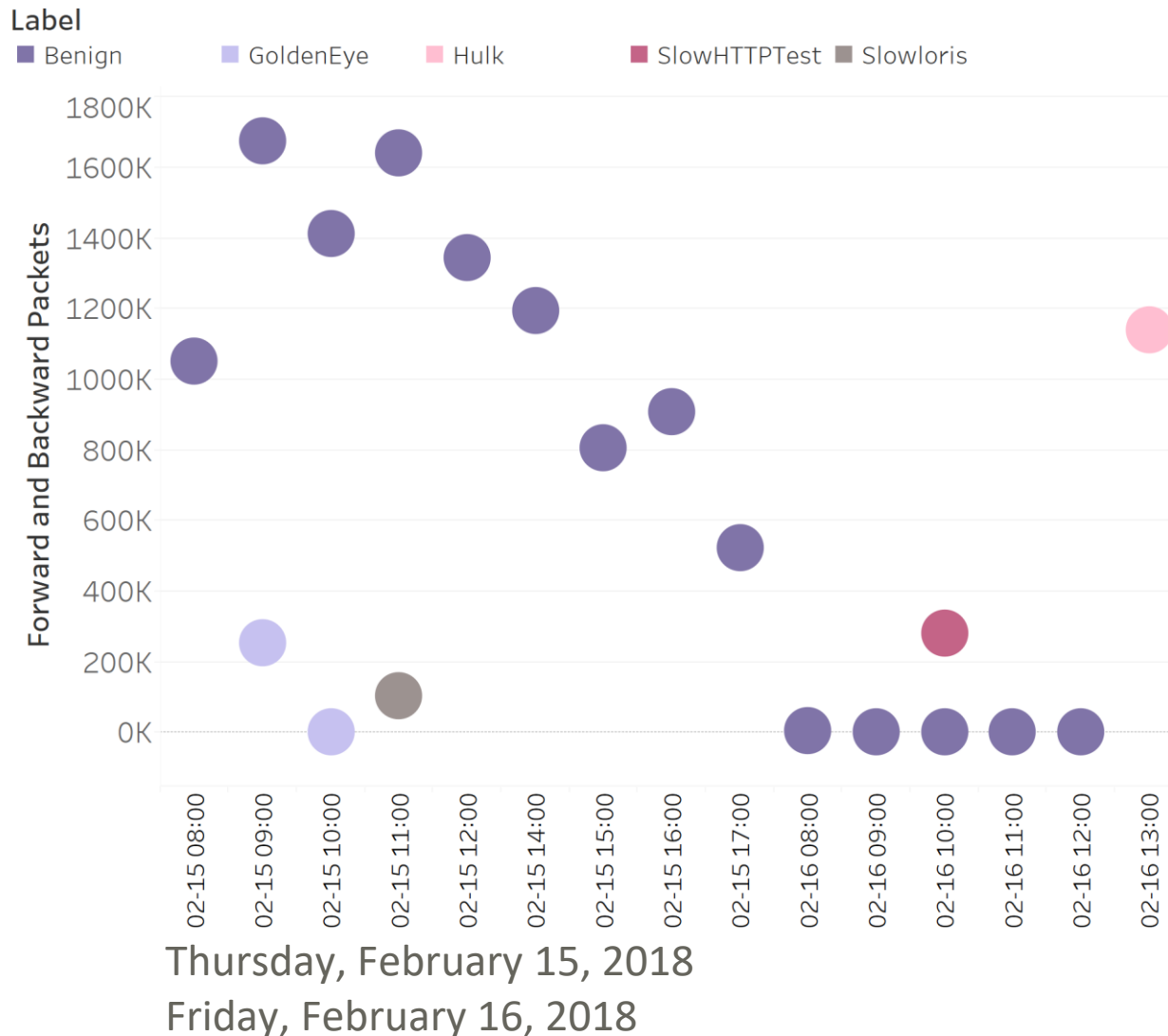
Intrusion Detection Evaluation datasets. [Online]. Available: <https://www.unb.ca/cic/datasets.html>.

CIC-IDS2017



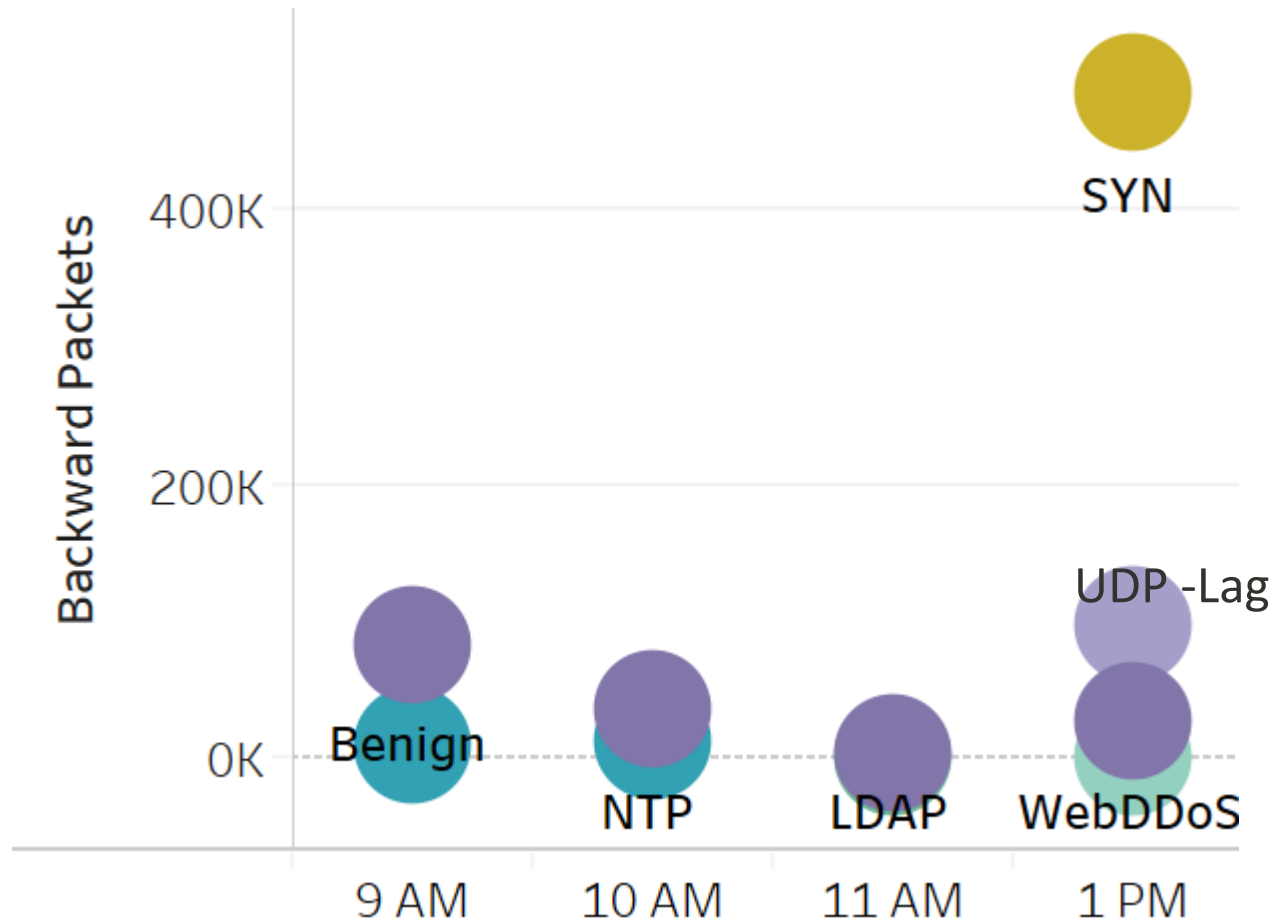
- **Attacker network:** one router, one switch, and four terminals
- **Victim-network:** three servers, one firewall, two switches, and ten terminals interconnected by a security authentication server.
- Monday July 3 - Friday July 7, 2017
- Patator, Slowloris, Heartleech, Damn Vulnerable Web App, Metasploit, Ares, and Low Orbit Ion Cannon

CSE-CIC-IDS2018



- **Attacker network:** 420 terminals and 30 servers split into 5 subnets
- **Victim-network:** 50 terminals implemented using Amazon Web Services
- 10 days: Wednesday, February 14 -and Friday, March 2, 2018
- **Attack scenarios:** Botnet, Brute-force, DoS, DDoS, Heartbleed, network infiltration, and Web attacks

CIC-DDoS2019

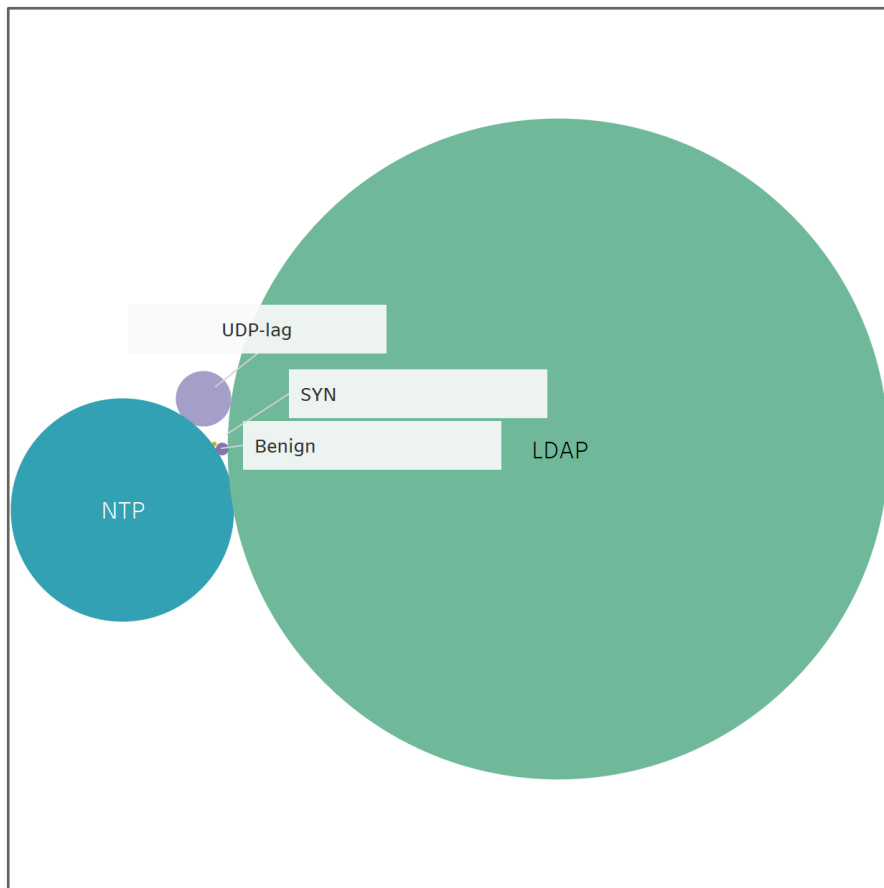


Saturday, January 12, 2019

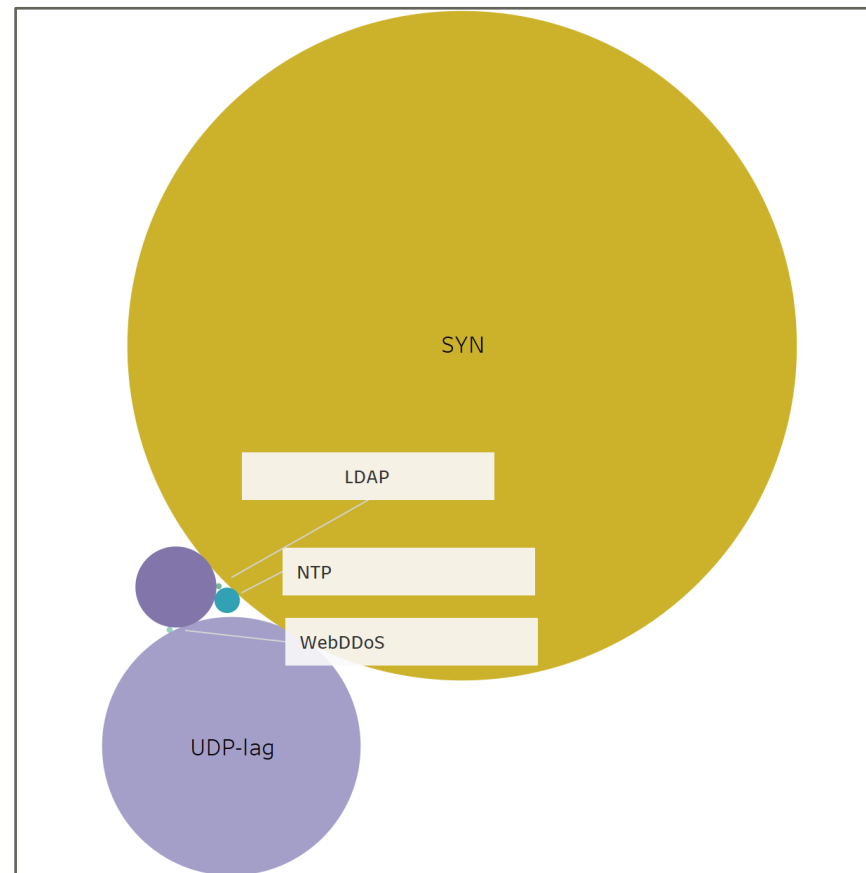
- **Attacker network:** third party infrastructure
- **Victim-network:** one Ubuntu 16 webserver, four personal computers, one Fortinet firewall, and two switches
- January 12, 2019, 10:30 -17:15:12 attacks: **NTP***, DNS, **LDAP***, MSSQL, NetBIOS, SNMP, SSDP, UDP, **UDP-Lag**, **WebDDoS**, SYN and TFTP.
- March 11, 2019, 9:40 - 17:35: 7 attacks: PortScan, NetBIOS, LDAP, MSSQL, UDP, UDPLag and SYN.

- NTP – Network Time Protocol;
- LDAP – Lightweight Directory Access Protocol

CIC-DDoS2019

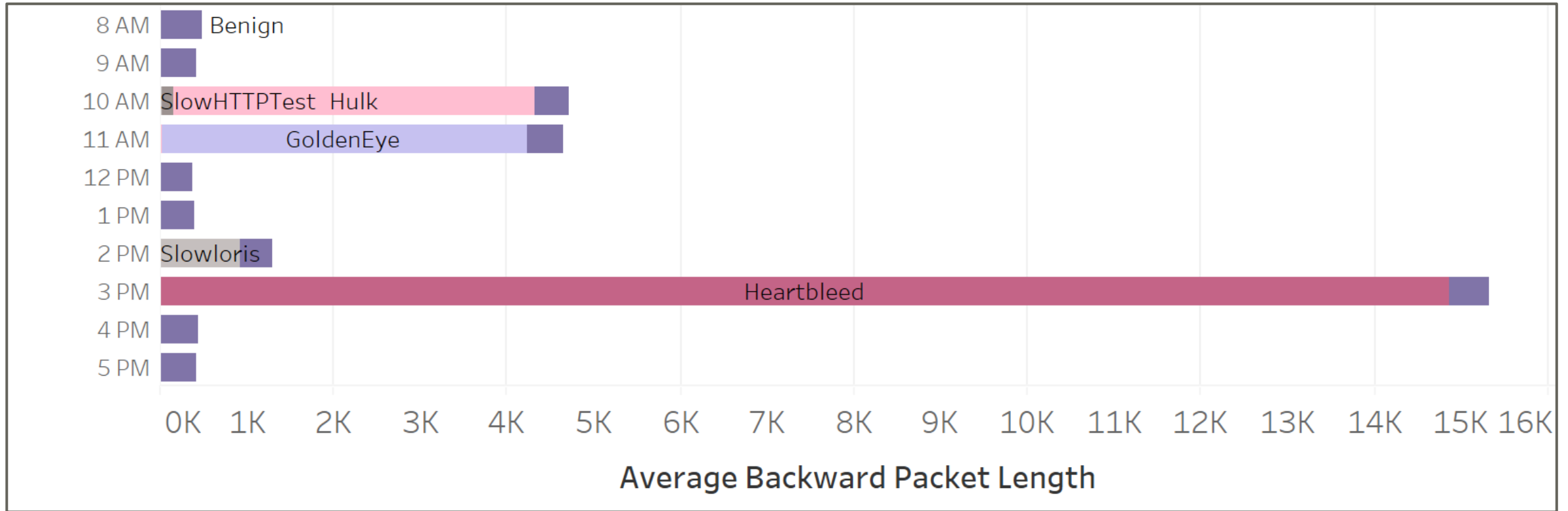


Average packet size



Average flow duration

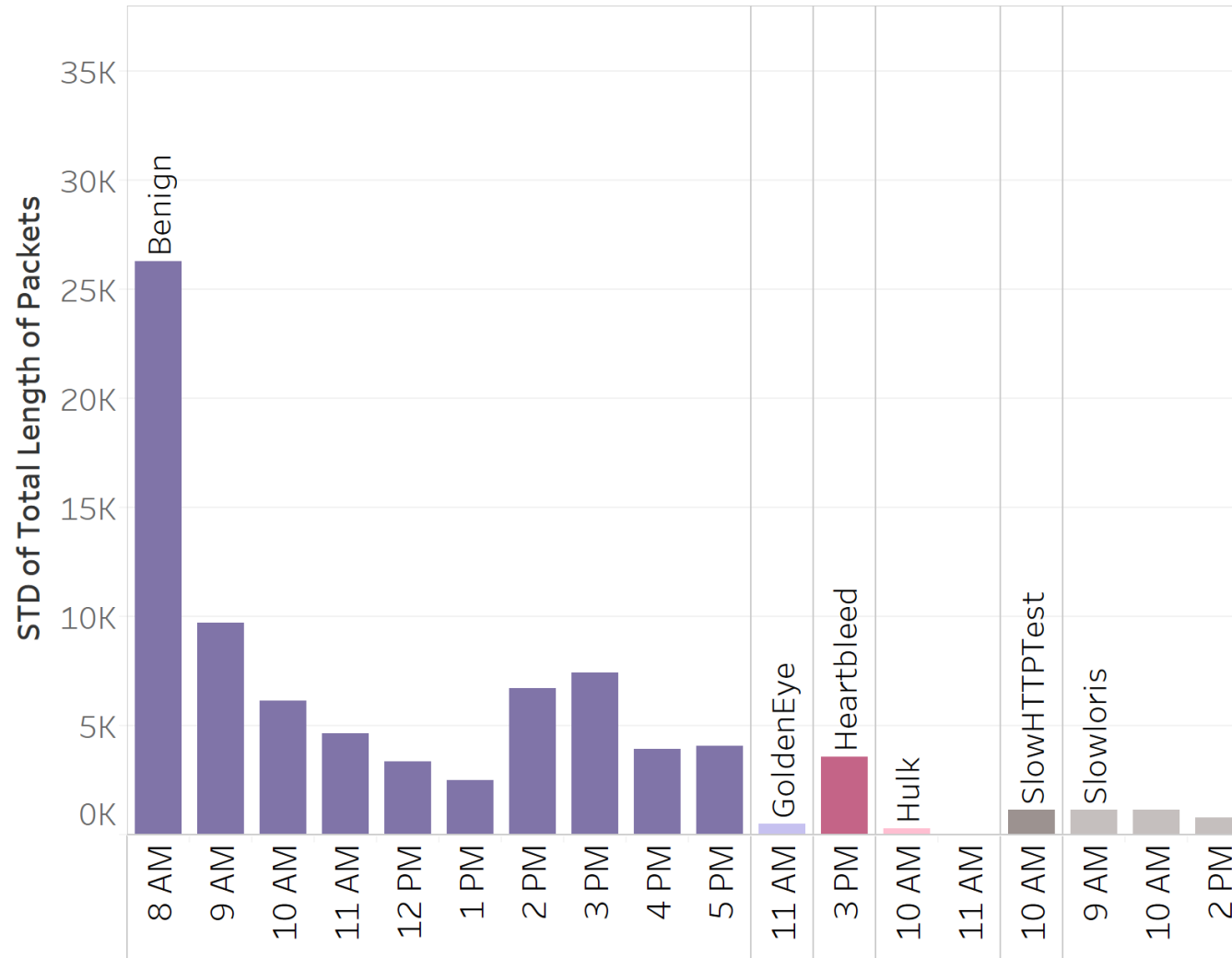
Features



Packet length (CIC-IDS2017):

- Regular packets are generally under 1,000 bytes
- Heartbleed attack packets approximately reach 15,000 bytes on average.

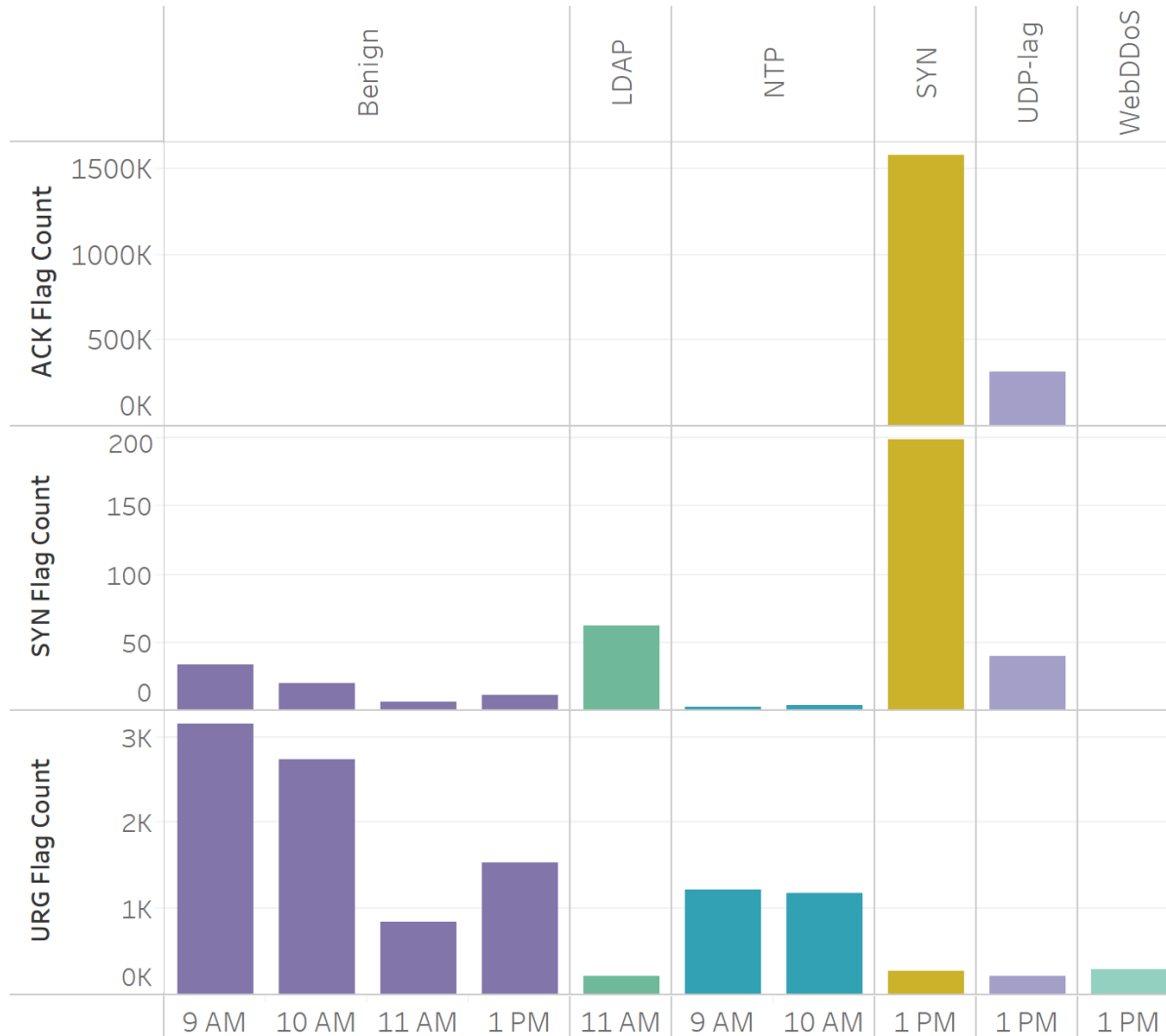
Features



Standard deviation of total length of packets CIC-IDS2017.

Regular packets generally have high length variation

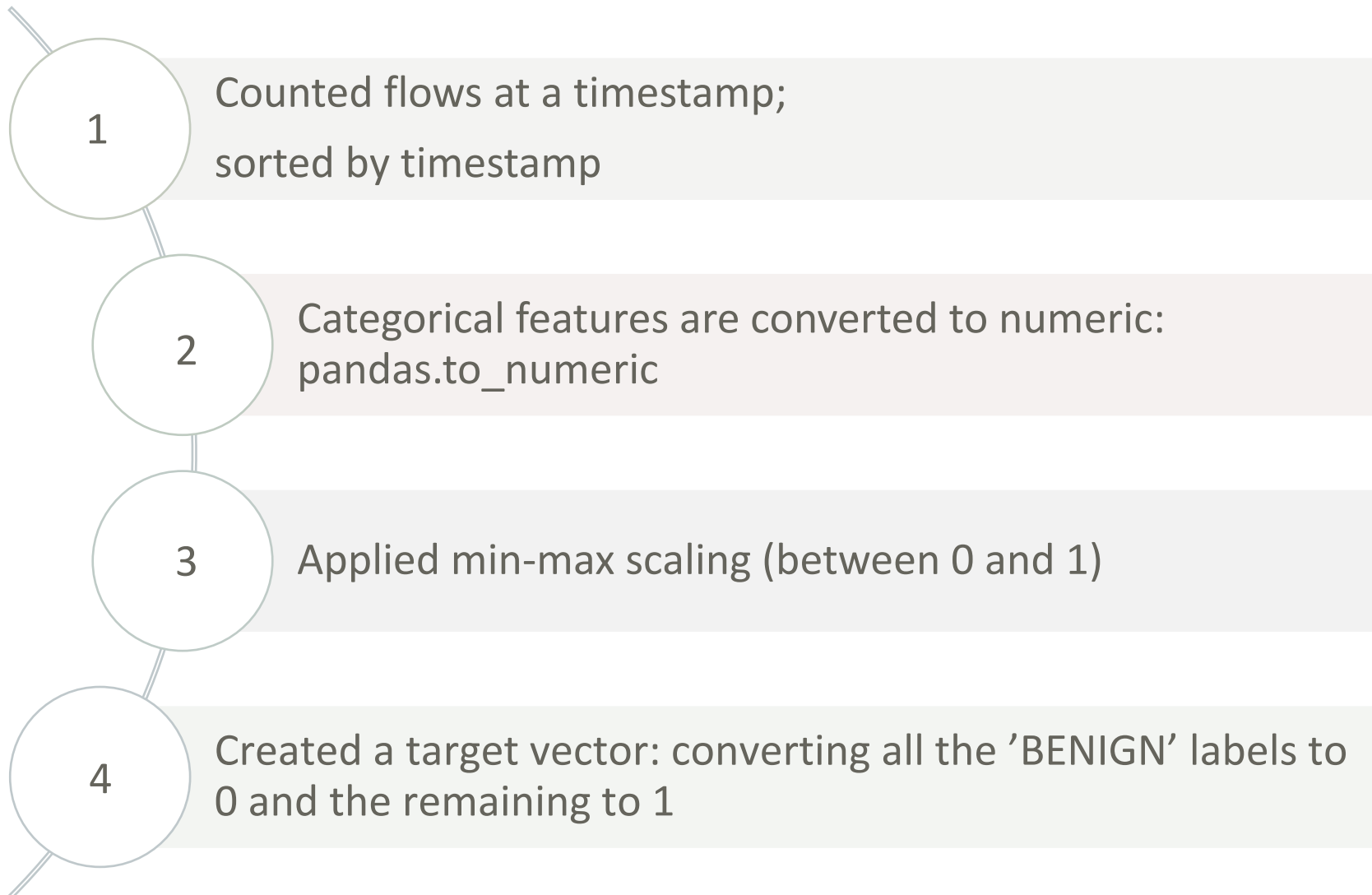
Features



TCP Flags (CIC-DDoS2019):

- SYN attacker brings down a network connection by requesting for seemingly legitimate connections through a series of TCP requests with TCP SYN, ACK flags set to 1

Data Preprocessing: CIC-IDS2017, CSE-CIC-IDS2018, and CICDDoS2019



Data Preprocessing: CIC-IDS2017, CSE-CIC-IDS2018, and CICDDoS2019

- Number of features: CIC-IDS2017 = **84**, CSE-CIC-IDS2018 = **79**, and CICDDoS2019= **85**
- Number of data points **before processing steps**:

Dataset	Class	Number of data points
CIC-IDS2017, Wednesday, July 5	Total	692,703
	Regular	440,031
	Anomaly	252,672
CSE-CIC-IDS2018, Thursday, February 15	Total	1,048,575
	Regular	996,077
	Anomaly	52,498
CSE-CIC-IDS2018, Friday, February 16	Total	1,048,575
	Regular	446,772
	Anomaly	601,802
CIC-DDoS2019, Saturday, January 12	Total	1,000,000
	Regular	3,654
	Anomaly	996,346

Data Preprocessing: CIC-IDS2017, CSE-CIC-IDS2018, and CICDDoS2019 Resampling to observe the effect of imbalanced datasets with ESNs

- Drawback of undersampling: valuable information may be removed
- “Naive resampling” - nothing about the data is assumed; simple to implement and fast to execute even with large and complex

Dataset	Class	After Oversampling	After Undersampling
CIC-IDS2017, Wednesday, July 5	Total Regular Anomaly	800,000 399,919 400,081	505,344 252,672 252,672
CSE-CIC-IDS2018, Thursday, February 15	Total Regular Anomaly	1,000,000 500,120 499,880	104,996 52,498 52,498
CIC-DDoS2019, Saturday, January 12	Total Regular Anomaly	1,000,000 500,153 499,847	Only 3,654 anomalies – no undersampling

Border Gateway Protocol Datasets

BGP

- Routing protocol
- Allows Autonomous Systems (ASes) exchange reachability information
- Incremental

Types of messages

- open
- update
- keep alive
- notification

BGP collectors

- RIPE (rrc04, Geneva; rrc14, Palo Alto)
- Routeviews (routeviews4, Eugene Oregon)

RIPE NCC: RIPE Network Coordination Center. [Online]. Available: <http://www.ripe.net/data-tools/stats/ris/ris-raw-data>.

University of Oregon Route Views project. [Online]. Available: <http://www.routeviews.org>.

Border Gateway Protocol Datasets

Zebra-dump
parser

• MRT - ASCII

C# tool

• 37 features

Feature	Name	Category
1	Number of announcements	<i>volume</i>
2	Number of withdrawals	<i>volume</i>
3	Number of announced NLRI prefixes	<i>volume</i>
4	Number of withdrawn NLRI prefixes	<i>volume</i>
5	Average <i>AS-path</i> length	<i>AS-path</i>
6	Maximum <i>AS-path</i> length	<i>AS-path</i>
7	Average unique <i>AS-path</i> length	<i>AS-path</i>
8	Number of duplicate announcements	<i>volume</i>
9	Number of implicit withdrawals	<i>volume</i>
10	Number of duplicate withdrawals	<i>volume</i>
11	Maximum edit distance	<i>AS-path</i>
12	Arrival rate	<i>AS-path</i>
13	Average edit distance	<i>volume</i>
14 – 23	Maximum <i>AS-path</i> length, where $n = (11, \dots, 20)$	<i>AS-path</i>
24 – 33	Maximum edit distance = n , where $n = (7, \dots, 16)$	<i>AS-path</i>
34	Number of Interior Gateway Protocol (IGP) packets	<i>volume</i>
35	Number of Exterior Gateway Protocol (EGP) packets	<i>volume</i>
36	Number of incomplete packets	<i>volume</i>
37	Packet size (B)	<i>volume</i>

Border Gateway Protocol Datasets

Event	Beginning	Duration (min)
Slammer	25.01.2003	869
Nimda	18.09.2001	1301
Code Red I	19.07.2001	600
DDoS 2019	22.10.2019	8 hours
DDoS 2020	17.02.2020	3 days

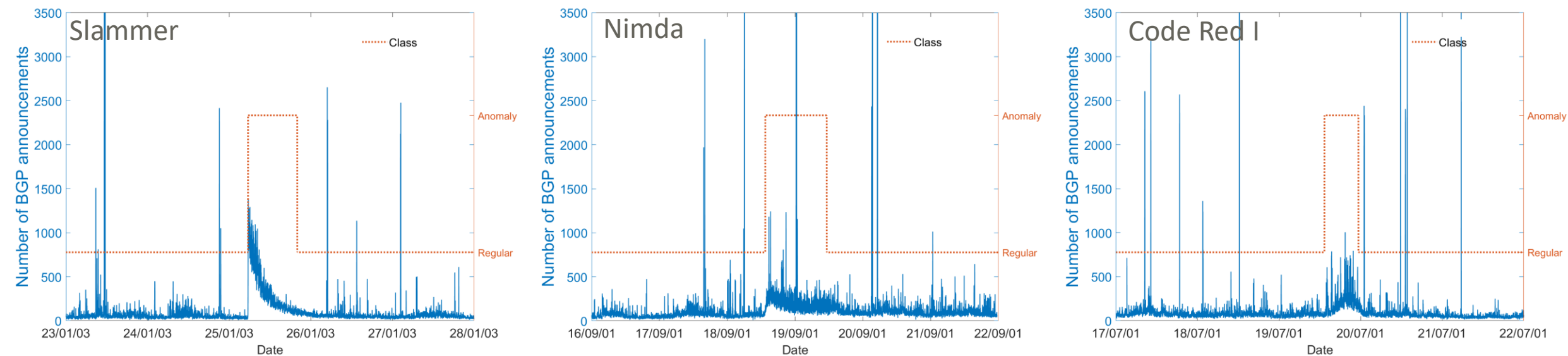
- BGP worms propagated via email messages
- DoS

Border Gateway Protocol Datasets

Event	Beginning	Duration (min)
Slammer	25.01.2003	869
Nimda	18.09.2001	1301
Code Red I	19.07.2001	600
DDoS2019	22.10.2019	8 hours
DDoS2020	17.02.2020	3 days

- **DDoS2019: October 2019 DDoS Attack on AWS:** affected the Amazon route 53 DNS webservice leaving thousands of customers not being able to access cloud services, websites, and applications.
- **DDoS2020: February 2020 DDoS Attack on AWS:** largest ever DDoS attack of 2.3 Tbps, CLDAP reflection attack.

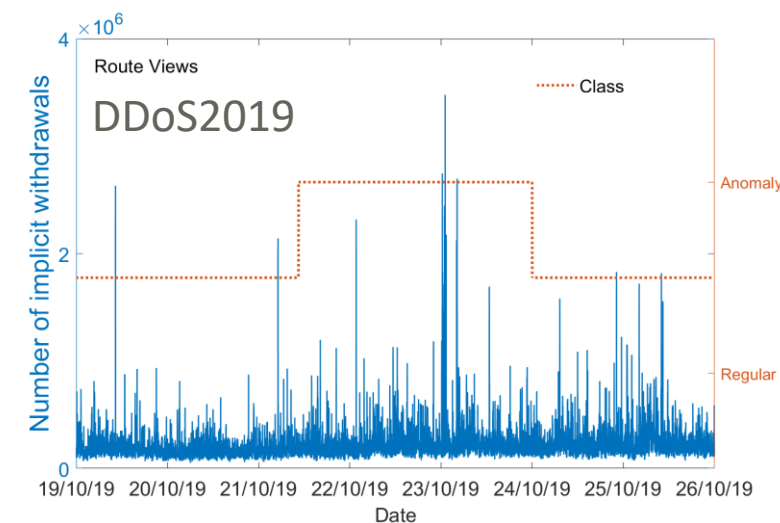
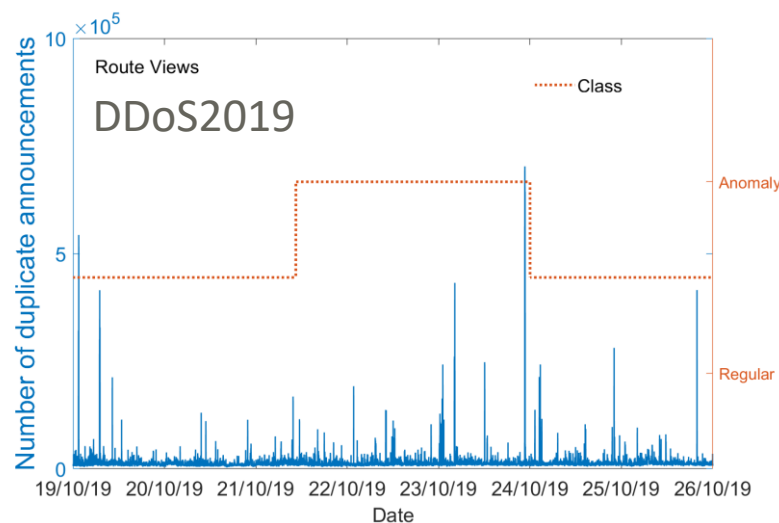
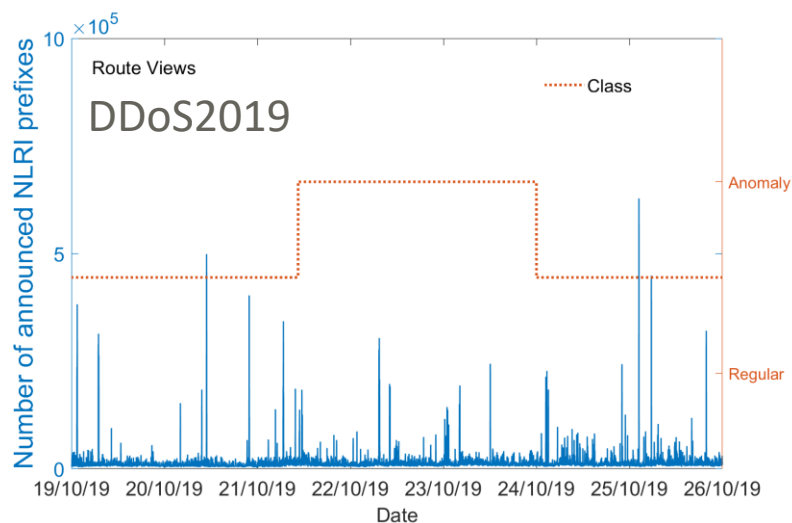
RIPE: Slammer, Nimda, Code Red I - Feature 1



Number of BGP announcements

Slammer (left), Nimda (center), and Code Red I (right).
The red dotted line indicates the class.

Route Views: October 2019 DDoS Attack on AWS



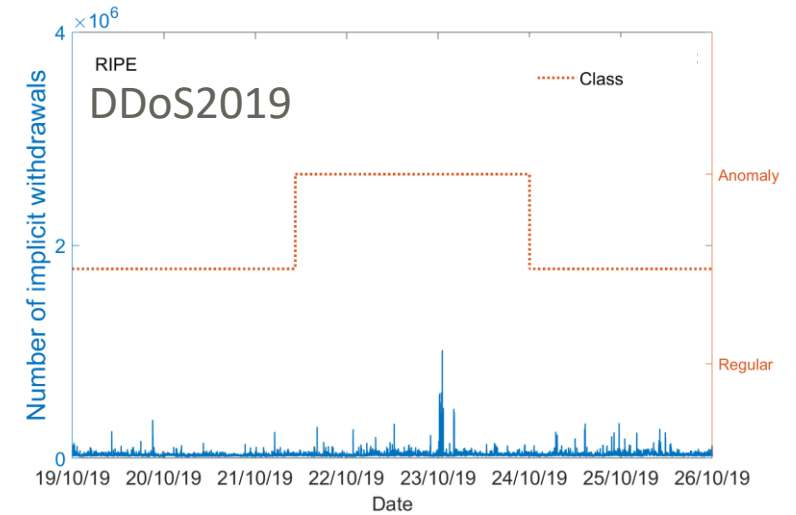
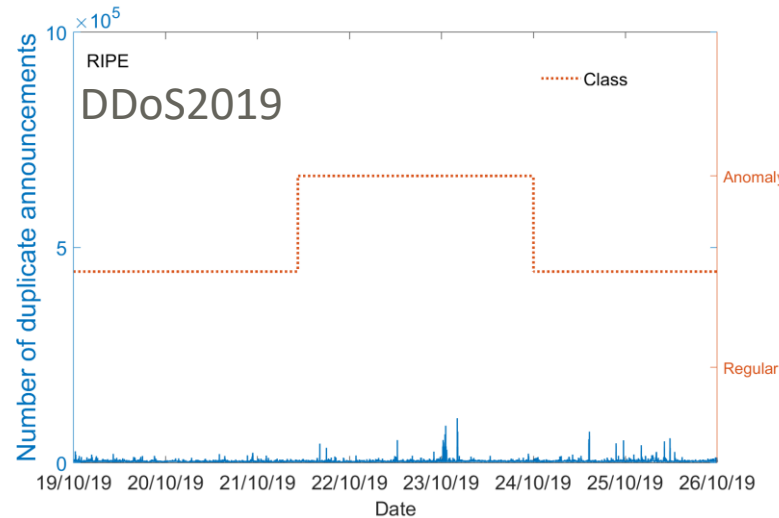
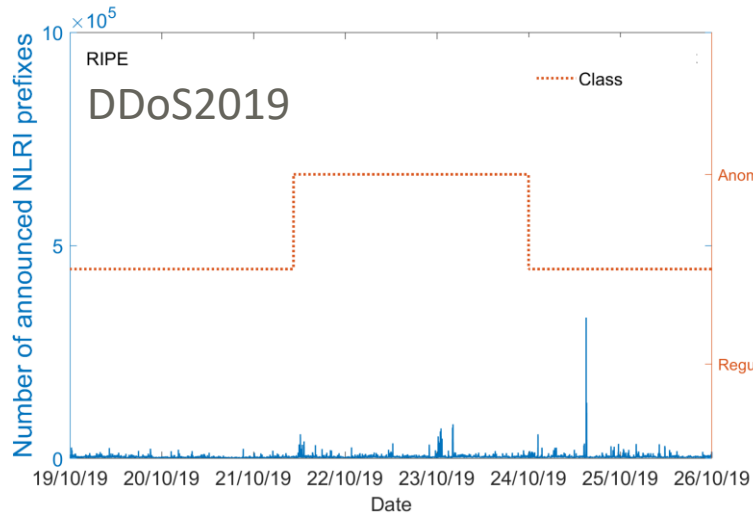
Number of announced NLRI* prefixes (left), number of duplicate announcements (center), and number of implicit withdrawals (right)

- Duplicate announcements are the BGP update packets that have identical NLRI prefixes and the AS-path attributes.
- Implicit withdrawals are prefixes implicitly withdrawn by sending the same prefix with new attributes.

We indicated the 23rd of October, 2019 as a day with network anomalies due to ransom driven DDoS attacks that hit the banking industry in South Africa

*NLRI – Near Layer Reachability Information

RIPE: October 2019 DDoS Attack on AWS

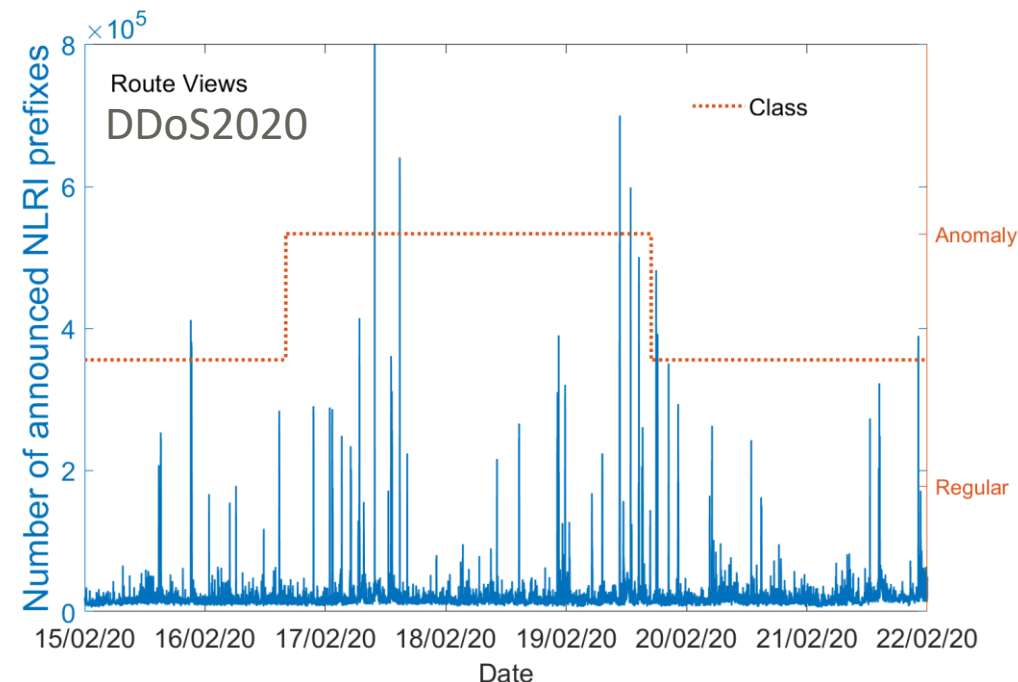
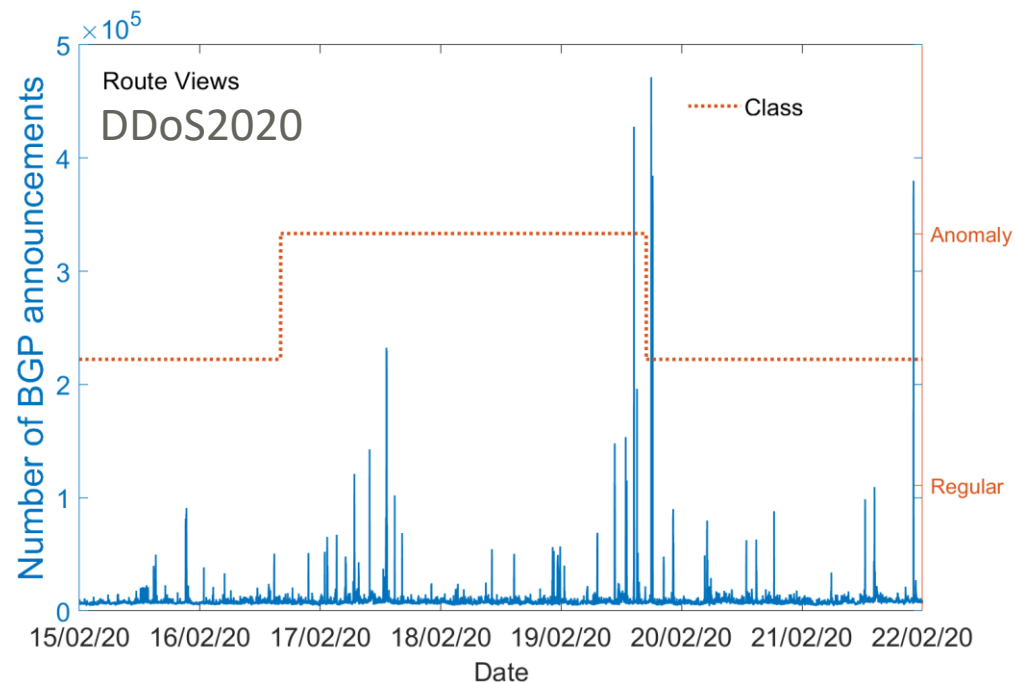


Number of announced NLRI prefixes (left), number of duplicate announcements (center), and number of implicit withdrawals (right)

- Duplicate announcements are the BGP update packets that have identical NLRI prefixes and the AS-path attributes.
- Implicit withdrawals are prefixes implicitly withdrawn by sending the same prefix with new attributes.

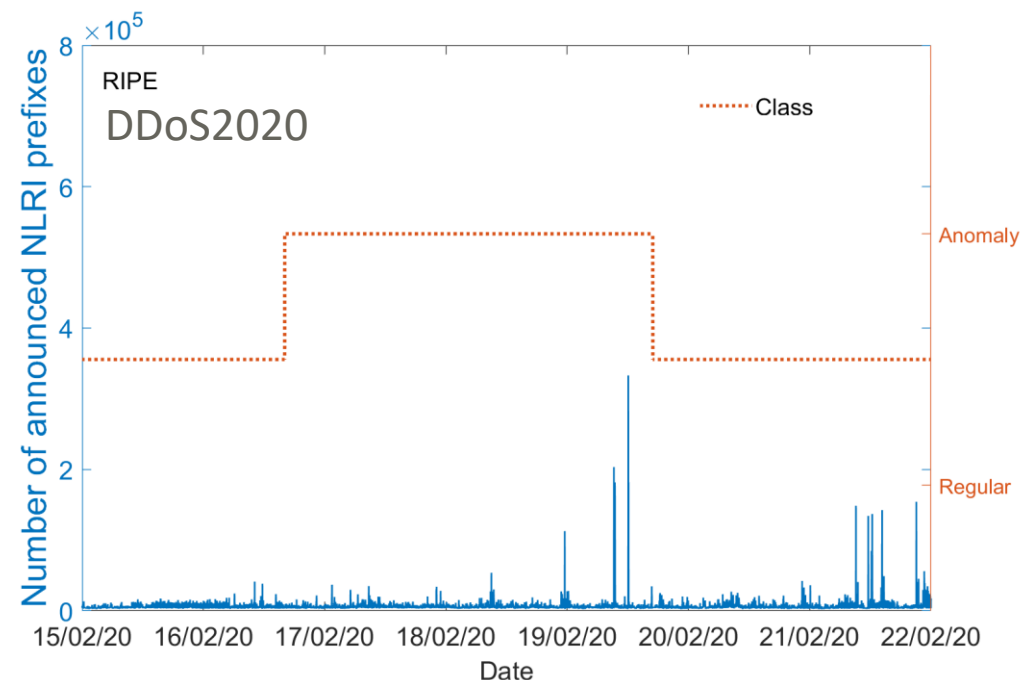
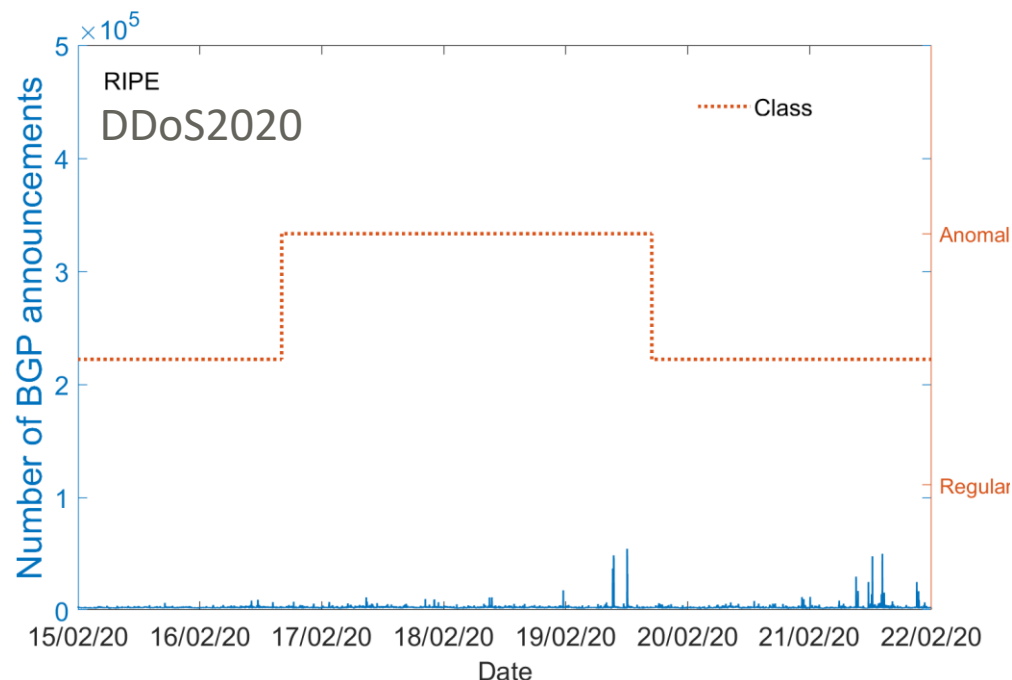
We indicated the 23rd of October, 2019 as a day with network anomalies due to ransom driven DDoS attacks that hit the banking industry in South Africa

Route Views: February 2020 DDoS Attack on AWS



Number of BGP announcements (left) and announced prefixes (right)

RIPE: February 2020 DDoS Attack on AWS



Number of BGP announcements (left) and announced prefixes (right)

Even though the attack lasted three days, we are able to see higher occurrences of the BGP updates starting February 21, 2020, which may influence the training of a machine learning model.

Data Preprocessing: BGP Datasets

Dataset	Class	Number of data points
Slammer	Total	7,200
	Regular	6,331
	Anomaly	869
Nimda	Total	8,609
	Regular	7,308
	Anomaly	1,301
Code Red I	Total	7,200
	Regular	6,600
	Anomaly	600
DDoS2019_v1	Total	7,200
	Regular	6,719
	Anomaly	481
DDoS2019_v2	Total	10,080
	Regular	6,390
	Anomaly	3,960
DDoS2020	Total	10,080
	Regular	5,709
	Anomaly	4,371

- No oversampling:

causes significant increase in the performance implying that machine learning models learn too well when smaller datasets get resampled

Feature Selection

Selecting
best features

- Enhances performance
- Reduces training time

Purpose

- Identify relevant features with the preservation of important discriminatory information

Decision
trees

- Posing conditions on a given point. A simple condition may be of the form: “Is feature i less than the value v ?”
- How to select split? When to stop growing trees?

Feature Selection: Extra Trees

Extra trees

- Extremely Randomized Trees
- Tree-based ensemble method generates decision trees from a training set.

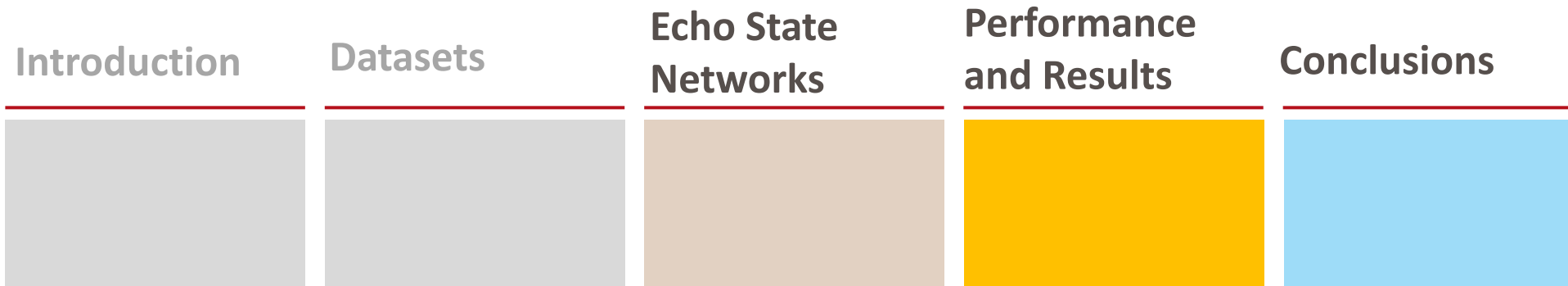
Ensemble learning

- Overcomes the overfitting by combining the predictions of many varied models into a single prediction

Parameters

- **Parameters:** number of attributes (features) (**K = 20**), minimum sample size (**nmin = 2**), number of decision trees in the ensemble (**M = 100**), determines the strength of the variance reduction of the ensemble model aggregation.

Roadmap



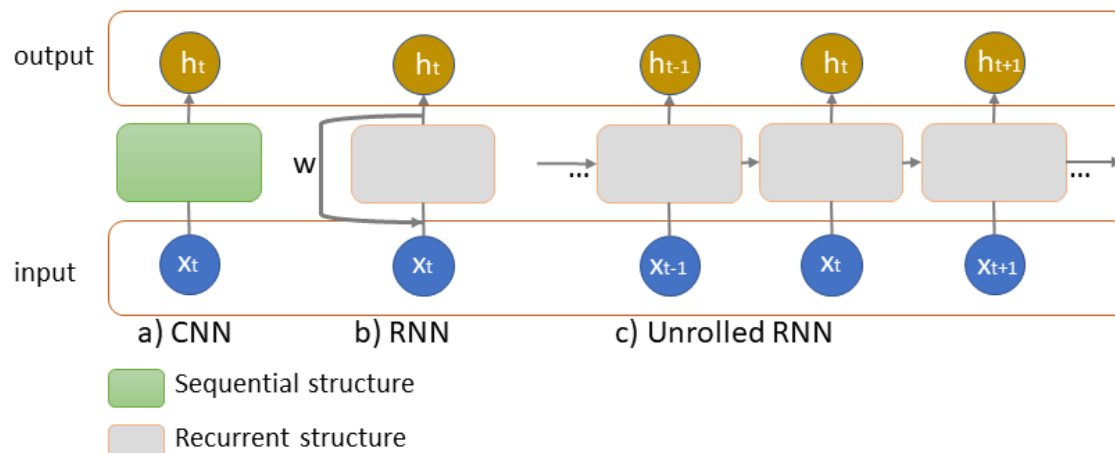
Roadmap

Echo State Networks

- Recurrent Neural Networks (RNNs)
- Reservoir Computing (RC) for training RNNs
- Echo State Networks (ESNs)
- ESN Reservoir Hyperparameters
- Cross-Validation in ESNs

Recurrent Neural Networks (RNNs)

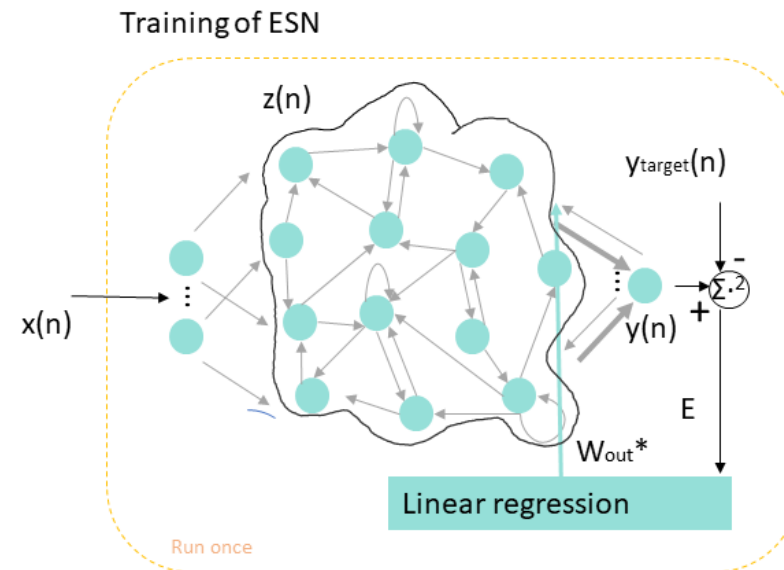
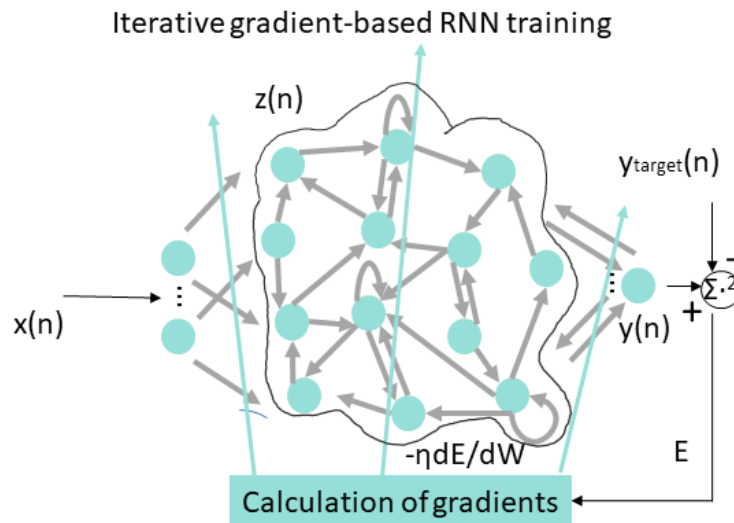
- RNNs belong to a class of artificial neural networks. They are widely used to detect anomalies in time-series datasets.
 - At each step the input of RNN is coming from a previous state of a hidden layer and is fed into the next state along with the new input signal. Often employ back propagation algorithm for training.



- Vanilla RNN: vanishing and exploding gradients;
- Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) cope with long-term dependencies.

Reservoir Computing (RC) as a Paradigm for Training RNNs

- Reservoir is a randomly connected network of nodes excited by input $x(n)$
- Reservoir weights are not changed by training
- Most common reservoirs are ESN and liquid state machine (LSM*): train only the memoryless output weights leaving out the supervised adaptation of input and reservoir weights.



*LSM is sparse neural network where activation functions are replaced by threshold levels. Reservoir accumulates values from sequential samples, and emits output only when the threshold is reached, setting internal counter again to zero.

Reservoir Computing (RC) as a Paradigm for Training RNN

$$z(n) = f(x(n)W^{in} + z(n-1)W + y(n-1)W^{fb}) \quad n = 1, \dots, N.$$

$z(n) \in R^{N_z}$ vector of reservoir activations at a timestep n

$f(\cdot)$ activation function ($\tanh(\cdot)$, applied elementwise)

$W^{in} \in R^{N_x \times N_z}$ randomly generated input weight matrix

$W \in R^{N_z \times N_z}$ randomly generated reservoir weight matrix

$W^{fb} \in R^{N_z \times N_y}$ optional output feedback weight matrix

$y(n) \in R^{N_y}$ output of the network:

$$y(n) = g([z(n); x(n)]W^{out}) \quad n = 1, \dots, N.$$

$W^{out} \in R^{N_{x+z} \times N_y}$ learned output weight matrix

$g(\cdot)$ output activation function ($\tanh(\cdot)$, applied elementwise, or identity function when using regression)

Training input sequence $x(n)$ is fed to the reservoir with reservoir's initial state equal to zero. The reservoir states are collected during training. Output weights are derived as the linear regression weights using target output.

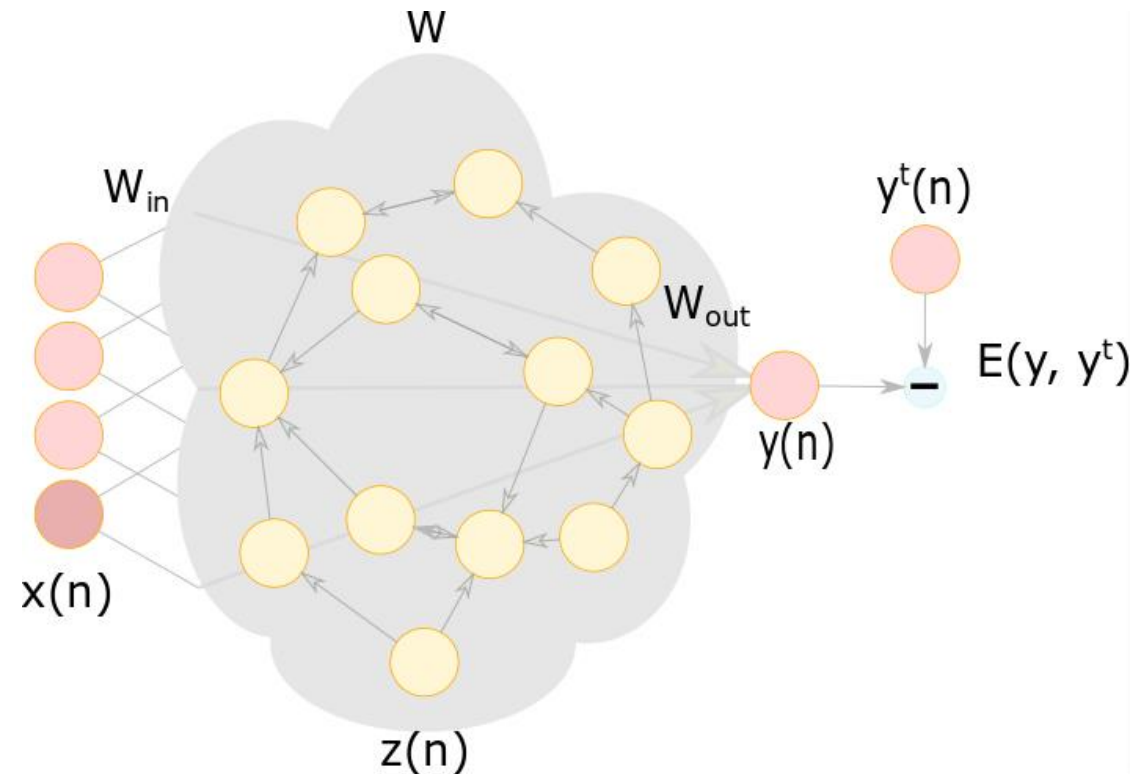
M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir Computing Trends", KI. Künstliche Intelligenz (Oldenbourg), vol. 26, no. 4, pp. 365–371, Nov. 2012.

Echo State Networks (ESN): Description

- ESNs: Conceptually and computationally simple to implement RC approach to train RNNs

Reservoir in ESN:

- Memory for the input $x(n)$
- Nonlinear high-dimensional expansion $z(n)$ of the input $x(n)$
- Characterized by tuple (W^{in}, W, α)



M. Lukosevicius, "A practical guide to applying Echo State Networks," in Neural Networks: Tricks of the Trade (2nd ed.), G. Montavon, G. B. Orr, and K. -R. Müller, Eds., Berlin, Heidelberg, Springer, 2012, vol. 7700, pp. 659–686.

ESN Reservoir Hyperparameters

Hyperparameters	Key Points
Size of the reservoir N_z	the larger number of nodes N_z in reservoir the better the performance (if proper regularization against overfitting is applied)
Sparsity of the reservoir	the sparser the connections (when most elements in \mathbf{W} are 0), the faster reservoir updates
Distribution of nonzero elements	nonzero element of \mathbf{W} (typically sparse matrix) and \mathbf{W}^{in} (typically dense matrix) have either symmetrical uniform, discrete bi-valued, or normal distribution centered around 0

ESN Reservoir Hyperparameters

Hyperparameters	Key Points
Spectral radius $\rho(\mathbf{W})$ (maximal eigenvalue of \mathbf{W})	spectral radius $\rho(\mathbf{W})$ defines how fast the influence of input dies out in reservoir with time (e.g., the larger the radius, the longer the memory of the input)
Input scaling	input scaling determines the amount of nonlinearity of $z(n)$ and the influence of the input on $z(n)$ as opposed to the history of the input normalize the data in order to keep the inputs bounded and avoid outliers (e.g., apply $\tanh(\cdot)$)
Leaking rate α	usually small dynamics of the reservoir extends the duration of the memory in ESN (usually tuned by trial and error)

ESN Models

	Deterministic	ρ	α	N_z
ESN1	False	0.9	0.2	10
ESN2	True	0.9	0.2	10
ESN3	False	0.1	0.2	10
ESN4	False	0.9	None	10
ESN5	False	0.9	0.2	30

- Deterministic reservoir - with each weight having the same value; known as recursive mechanism.
- ρ – reservoir radius
- α – leaking rate
- N_z – number of reservoir nodes

ESNs: Description (Steps)

Step 1: Generating random reservoir with parameters: $\mathbf{W}^{in} \in R^{N_x \times N_z}$, $\mathbf{W} \in R^{N_z \times N_z}$, $\alpha \in (0,1]$ – leaking rate

Step 2: Calculating reservoir activation states $\tilde{z}(n) \in R^{N_z}$ from the training set.

$$\begin{aligned}\tilde{z}(n) &= \tanh(x(n)\mathbf{W}^{in} + z(n-1)\mathbf{W}) \quad n = 1, \dots, N. \\ z(n) &= (1 - \alpha)z(n-1) + \alpha\tilde{z}(n) \quad n = 1, \dots, N.\end{aligned}$$

$\tilde{z}(n) \in R^{N_z}$ vector of reservoir neuron activations at a timestep n

$z(n) \in R^{N_z}$ the reservoir state update at a timestep n . N_z is a number of reservoir nodes

In cases where $\alpha = 1$ and $z(n) \equiv \tilde{z}(n)$.

ESN: Description (Steps)

Step 3: Using linear regression to obtain the output weights.

The vectors $[z(n); x(n)]^T$ are collected into a matrix $Z \in \mathbb{R}^{N \times (N_z + N_x)}$. Targets $y^{\text{target}}(n) \in \mathbb{R}^1$ are collected into a matrix $Y \in \mathbb{R}^{N \times 1}$. Z and Y have a row for every training time step n

$$\mathbf{W}^{\text{out}} = \mathbf{Z}^\dagger \mathbf{Y}$$

To find the optimal weights – we minimize the loss function:

$$E(\mathbf{y}, \mathbf{y}^{\text{target}}) = \frac{1}{N_y} \sum_{n=1}^{N_y} \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i(n) - y_i^{\text{target}}(n))^2}.$$

Step 4: Evaluating the network by applying collected output weights with the new input $x(n)$ to compute $y(n)$

$$y(n) = [z(n); x(n)] \mathbf{W}^{\text{out}} \quad n = 1, \dots, N.$$

$\mathbf{W}^{\text{out}} \in \mathbb{R}^{(N_z + N_x) \times 1}$ *learned output weight matrix*

ESN: Training

- Ridge regression is used to learn optimal output weights \mathbf{W}^{out} :

$$\mathbf{W}^{out} = (\mathbf{Z}^T \mathbf{Z} + \beta \mathbf{I})^{-1} \mathbf{Z}^T \mathbf{Y}^{target}$$
$$\mathbf{W}^{out} = \underset{\mathbf{W}^{out}}{\operatorname{argmin}} \frac{1}{N_Y} \sum_{i=1}^{N_Y} \left(\sum_{n=1}^N y_i^{target}(n) - y_i(n) \right)^2 + \beta ||\mathbf{w}_i^{out}||^2$$

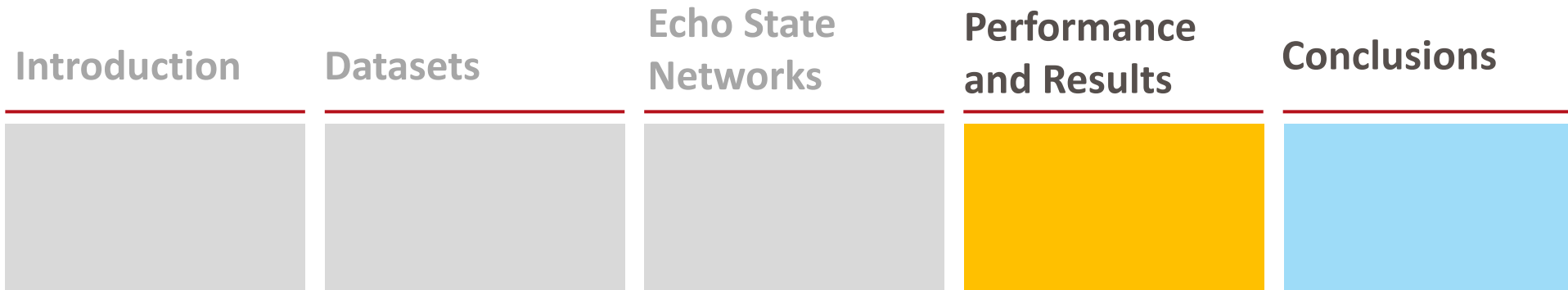
\mathbf{Y}^{target} and $\mathbf{Y} \in R^{N \times N_Y}$: label matrix and output matrix

Design matrix \mathbf{Z} is used instead of $[\mathbf{X}; \mathbf{Z}]$ for conciseness

β is a regularization coefficient and \mathbf{I} is the Identity matrix

- Including scaled white noise to the input can serve a similar purpose as regularization
- Smaller reservoir sizes, and/or shorter datasets speed up the training

Roadmap



Roadmap

Performance and Results

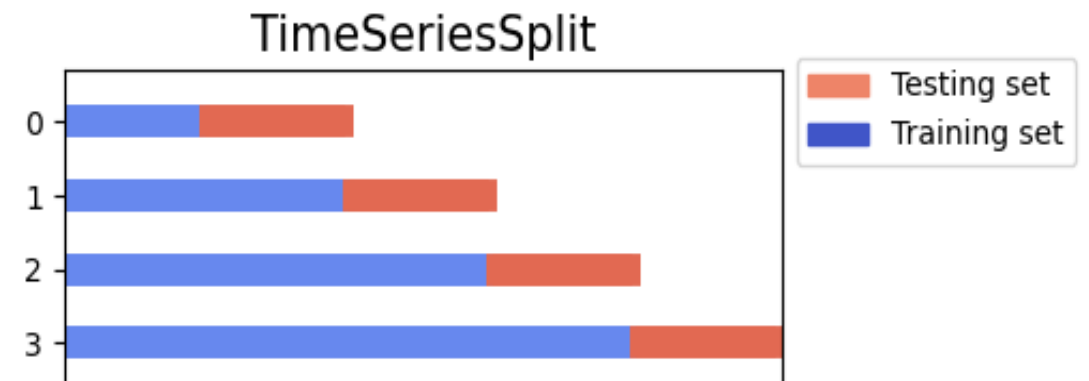
- Performance of ESN Models with Balanced and Unbalanced Datasets
- Comparing Performance of ESN and Bi-LSTM in Detecting the Denial of Service Attacks

Cross-Validation

- Choosing the hyperparameters is known as model selection.
- It is not recommended to select models using a test set. Instead, a training set is split into smaller subsets where validation subsets are used to evaluate the model.

K-fold cross-validation, the most widely used validation technique, allows $(K - 1)/K$ of the data to be used for training and the rest to assess performance.

- **Benefits:** May prevent overfitting, improve stability: training on multiple folds creates additional regularization. Averaging validation over many folds reduces the effects of occasional imperfections
- **Drawbacks:** The training set is smaller, so the validation loss is a less accurate gauge of true performance on the testing set.
Computationally intensive



Performance Results

CIC-IDS2017, Wednesday, July 5 2017

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.929	0.928	0.129	0.927	0.907	0.106
ESN2	0.927	0.925	0.136	0.958	0.945	0.058
ESN3	0.895	0.894	0.176	0.915	0.893	0.120
ESN4	0.900	0.899	0.189	0.919	0.899	0.120
ESN5	0.967	0.950	0.057	0.973	0.965	0.038

CIC-CSE-IDS2018, Friday, February 16 2018

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.988	0.990	0.115	0.997	0.998	0.006
ESN2	0.934	0.937	0.123	0.980	0.828	0.020
ESN3	0.985	0.988	0.117	0.996	0.996	0.010
ESN4	0.991	0.993	0.032	0.999	0.999	0.003
ESN5	0.995	0.996	0.009	0.999	0.999	0.000

CIC-CSE-IDS2018, Thursday, February 15 2018

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.938	0.937	0.109	0.983	0.854	0.017
ESN2	0.927	0.925	0.113	0.980	0.828	0.020
ESN3	0.798	0.787	0.401	0.961	0.679	0.032
ESN4	0.855	0.851	0.259	0.979	0.824	0.021
ESN5	0.945	0.944	0.011	0.997	0.973	0.003

CIC-DDoS2019, Saturday, January 12 2019

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.989	0.989	0.017	0.994	0.994	0.012
ESN2	0.992	0.991	0.013	0.999	0.997	0.000
ESN3	0.922	0.921	0.154	0.927	0.932	0.146
ESN4	0.957	0.957	0.077	0.981	0.999	0.000
ESN5	0.998	0.998	0.002	0.999	0.999	0.001

Performance of ESN models based on accuracy, F-Score, and false alarm rate when evaluated using **CIC-IDS2017** Wednesday, July 5 (**unbalanced**), **CIC-CSE-IDS2018** Thursday, February 15 (**unbalanced**), **CIC-CSE-IDS2018** Friday, February 16 (**balanced**), and **CIC-DDoS2019** Saturday, January 12 (**balanced**)

Performance Results: Oversampling and Undersampling

CIC-IDS2017, Wednesday, July 5 2017

	Oversampling			Undersampling		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.926	0.930	0.127	0.925	0.929	0.135
ESN2	0.946	0.948	0.099	0.920	0.924	0.140
ESN3	0.911	0.917	0.159	0.818	0.840	0.321
ESN4	0.896	0.906	0.202	0.924	0.928	0.133
ESN5	0.971	0.972	0.052	0.960	0.960	0.074

Performance of ESN models based on accuracy, F-Score, and false alarm rate when evaluated using **oversampled and undersampled CIC-IDS2017** and **CIC-CSE-IDS2018** Thursday

CIC-CSE-IDS2018, Thursday, February 15 2018

	Oversampling			Undersampling		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.981	0.982	0.035	0.970	0.971	0.059
ESN2	0.976	0.976	0.046	0.981	0.981	0.038
ESN3	0.891	0.902	0.215	0.837	0.860	0.322
ESN4	0.982	0.982	0.036	0.823	0.850	0.355
ESN5	0.990	0.991	0.018	0.988	0.989	0.022

Performance Results

Slammer

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.587	0.549	0.446	0.907	0.699	0.080
ESN2	0.625	0.654	0.366	0.908	0.710	0.083
ESN3	0.536	0.563	0.453	0.930	0.726	0.036
ESN4	0.505	0.524	0.471	0.927	0.712	0.036
ESN5	0.636	0.669	0.341	0.900	0.699	0.095

Performance of ESN models based on accuracy, F-Score, and false alarm rate when evaluated using

BGP datasets: Slammer, Nimda, Code Red I

Nimda

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.463	0.465	0.512	0.805	0.502	0.166
ESN2	0.507	0.529	0.473	0.821	0.470	0.130
ESN3	0.446	0.439	0.507	0.843	0.167	0.024
ESN4	0.436	0.433	0.514	0.841	0.122	0.021
ESN5	0.492	0.497	0.513	0.818	0.516	0.150

Code Red I

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.619	0.612	0.331	0.910	0.432	0.040
ESN2	0.636	0.671	0.358	0.919	0.424	0.027
ESN3	0.678	0.700	0.270	0.913	0.046	0.002
ESN4	0.907	0.876	0.001	0.901	0.536	0.075
ESN5	0.598	0.605	0.401	0.910	0.547	0.062

Performance Results

DDoS2019, RIPE

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.622	0.621	0.351	0.571	0.502	0.465
ESN2	0.618	0.623	0.389	0.579	0.558	0.527
ESN3	0.549	0.546	0.398	0.481	0.522	0.702
ESN4	0.564	0.552	0.399	0.525	0.505	1.000
ESN5	0.602	0.611	0.361	0.677	0.617	0.371

DDoS2020, RIPE

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.520	0.506	0.452	0.439	0.610	0.988
ESN2	0.529	0.491	0.400	0.437	0.606	0.994
ESN3	0.529	0.491	0.390	0.437	0.607	0.998
ESN4	0.513	0.512	0.453	0.436	0.607	1.000
ESN5	0.539	0.536	0.444	0.453	0.610	0.955

DDoS2019, Route Views

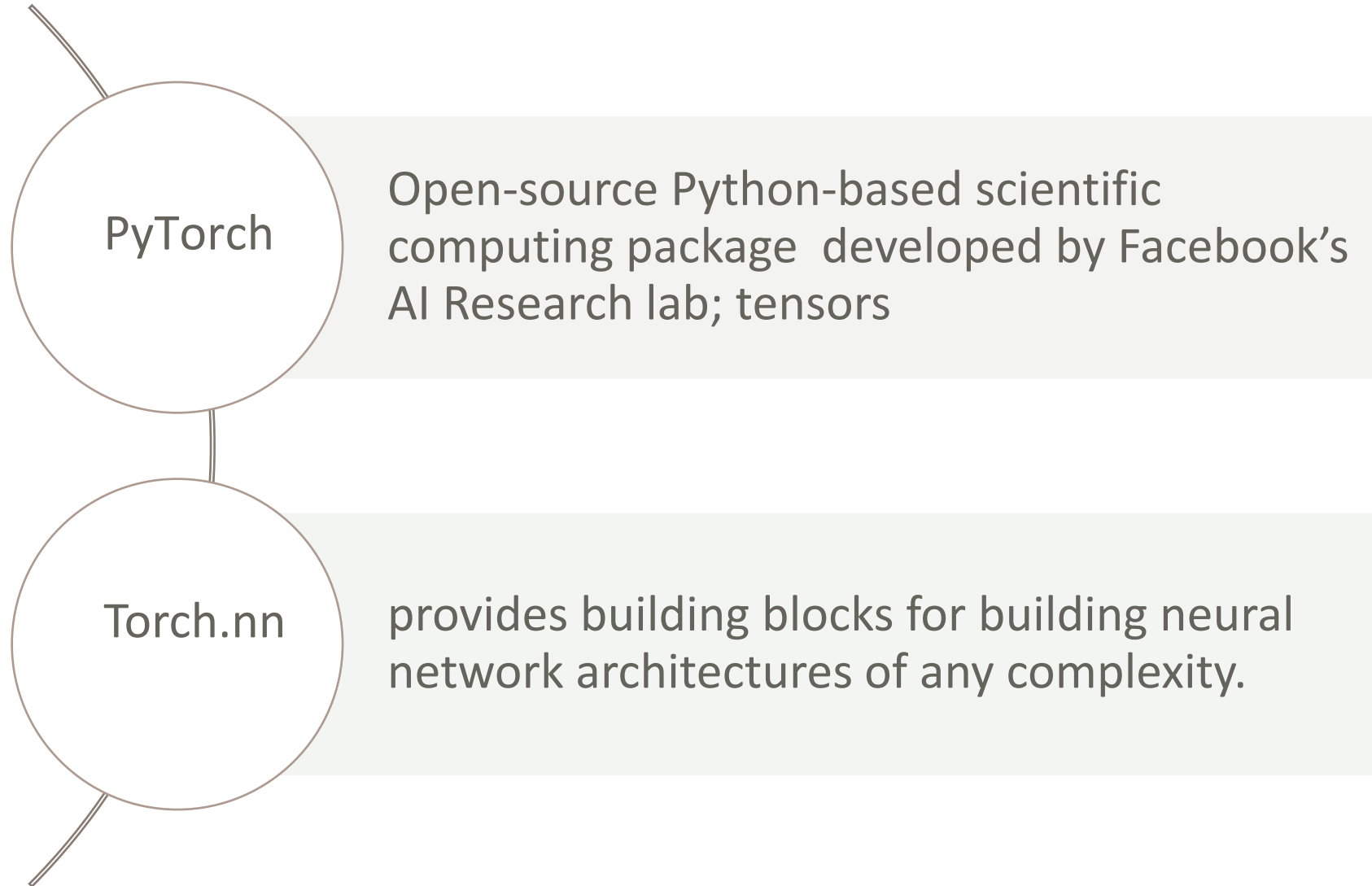
	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.560	0.528	0.378	0.613	0.433	0.259
ESN2	0.555	0.587	0.374	0.611	0.551	0.406
ESN3	0.551	0.552	0.394	0.615	0.261	0.130
ESN4	0.526	0.528	0.399	0.624	0.193	0.084
ESN5	0.590	0.659	0.350	0.618	0.540	0.373

DDoS2020, Route Views

	Validation			Test		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN1	0.513	0.491	0.400	0.477	0.609	0.877
ESN2	0.516	0.496	0.399	0.577	0.610	0.565
ESN3	0.508	0.483	0.410	0.437	0.603	0.982
ESN4	0.503	0.473	0.408	0.441	0.604	0.971
ESN5	0.553	0.554	0.413	0.595	0.621	0.536

Performance of ESN models based on accuracy, F-Score, and false alarm rate when evaluated using BGP datasets collected from **RIPE** and **Route Views**: **DDoS2019_v2** (left), and **DDoS2020** (right)

Performance Results: ESN and Bi-LSTM



Performance Results: ESN and Bi-LSTM

Bidirectional LSTM layer: input nodes = number of features and 16 output nodes, dropout rate = 0.5, batch size = 10, and ReLU activation function

Fully-connected layer with 2 output nodes

The last layer returns logits - raw values which are passed to the F.softmax module

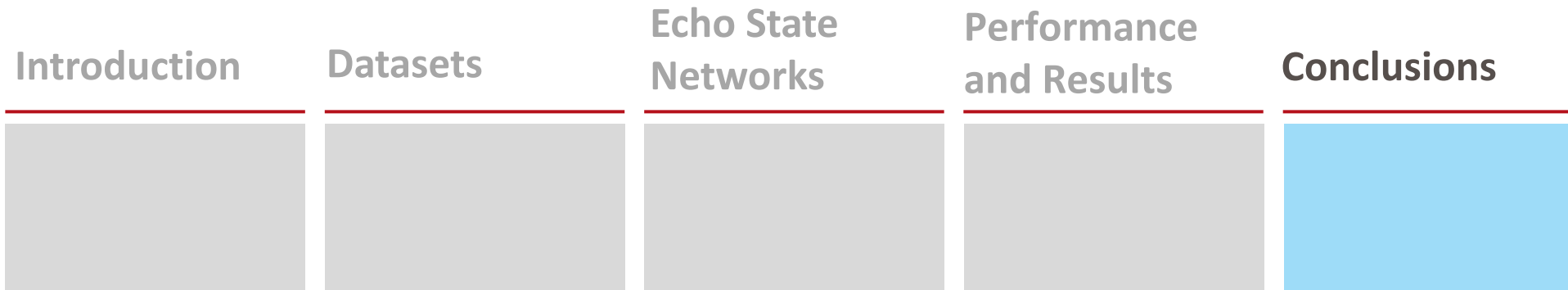
```
nn.CrossEntropyLoss() ; torch.optim.Adam(); learning rate 0.001
```


Performance Results: ESN and Bi-LSTM

	Bi-LSTM				ESN5			
	Acc.	F-Score	FAR	Time (s)	Acc.	F-Score	FAR	Time (s)
CIC-IDS Datasets:								
CIC-IDS2017	0.995	0.994	0.002	2,200	0.973	0.965	0.038	988
CSE-CIC-IDS2018, Thursday	0.996	0.962	0.004	3,417	0.997	0.973	0.003	2,335
CSE-CIC-IDS2018, Friday	0.976	0.979	0.000	3,149	0.999	0.999	0.000	2,369
CIC-DDoS2019	1.000	1.000	0.000	2,619	0.999	0.999	0.001	1,690
BGP Worm Datasets:								
Slammer	0.958	0.827	0.024	34	0.900	0.699	0.095	8
Nimda	0.863	0.375	0.029	41	0.818	0.516	0.150	7
Code Red I	0.929	0.491	0.021	37	0.910	0.547	0.062	6
BGP DDoS Datasets:								
DDoS2019, RIPE	0.388	0.478	0.837	111	0.677	0.617	0.371	12
DDoS2019, Route Views	0.654	0.791	1.000	99	0.618	0.540	0.373	6
DDoS2020, RIPE	0.346	0.514	1.000	107	0.453	0.610	0.955	9
DDoS2020, Route Views	0.760	0.864	1.000	101	0.595	0.621	0.536	11

- When evaluated using **CIC-IDS datasets** and **BGP Nimda and Code Red I datasets**, ESN and Bi-LSTM show comparable performance. When evaluated using **BGP Slammer dataset** and **BGP DDoS2019 and DDoS2020 Route Views datasets**, Bi-LSTM outperforms ESN. When evaluated using **BGP DDoS2019 and DDoS2020 RIPE datasets**, ESN slightly outperforms LSTM. The ESN **training time** is faster because ESN is not employing backpropagation.

Roadmap



Roadmap

Conclusions

- Conclusion and Future Work
- Key References

Conclusion and Future Work

- DoS and DDoS detection is becoming a challenging task due to changing network behavior
- In this Thesis we apply machine learning techniques to detect DoS and DDoS attacks and to show that **echo state networks (ESN) is a feasible method**.
 - We have selected **CIC-IDS2017, CSE-CIC-IDS2018, and CIC-DDoS2019**. **Synthetically generated datasets** contain regular data samples and randomly added artificial anomalies.
 - We also used **data from deployed networks** collected from a public repositories **Réseaux IP Européens (RIPE) and Route Views**. We observed how **recent large DDoS** attacks are reflected in BGP traffic records.
 - **ESN models' performance is better with CIC-IDS datasets than with BGP datasets:**
 - CIC-IDS: contain records of various protocols: HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP; the **variety of features** provide more information of regular and anomalous behavior for machine learning models.
 - BGP trace collectors may provide only **estimates of AS-level** Internet topologies and including additional data from route servers and looking glasses may help capture complete AS-level topology.
 - **Lack of ground truth: labeling of regular and anomaly** data as indicated periods of anomaly might contain the regular BGP records that are categorized as anomalous.
 - Different **size of datasets**: it's always better to train on larger datasets.

Conclusion and Future Work

- Selected **echo state networks models** and observed the effect of **hyperparameters**:
 - Increasing the **number of reservoir nodes** enhanced the performance of echo state networks
 - Decreasing **the radius of the reservoir** slightly degraded the performance
- Performance of echo state networks was evaluated with extracting the most **important features**: Selecting **relevant features** enhances classification results of ESN.
- **k-fold cross-validation** in ESN may be computationally less expensive no need to rerun the whole network when training the model.
- We compared echo state networks to **Bi-LSTM**: Both models showed **comparable** performance while the ESN training time was faster.

Conclusion and Future Work

- Echo state networks involve a degree of **uncertainty** in tuning some of the hyperparameters.
- Echo state networks had been shown to be a **feasible approach for network intrusion detection**.
- **Future Work with ESNs:**
 - Experiment with other **training algorithms** and **feedbacks** to improve efficiency.
 - Employ **bidirectional algorithm** that captures dependencies in the data forward and backward in time.
 - Use other existing real and synthetic **network datasets**.

Key References

DoS and DDoS Detection:

- E. Chou and R. Groves, Distributed Denial of Service (DDoS): Practical Detection and Defense. 1st Ed. Sebastopol, CA: O'Reilly Media, 2018.
- V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: a survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 15:1–15:58, July 2009.
- J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” ACM SIGCOMM Comput. Commun. Rev., 34, 2004, pp. 39–53.

Key References

Machine Learning:

- C. M. Bishop, Pattern Recognition and Machine Learning. Secaucus, NJ, USA: Springer-Verlag, 2006.
- I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: The MIT Press, 2016.
- M. Lamons, R. Kumar, and A. Nagaraja, Python Deep Learning Projects, Packt Publishing, 2018. [E-book] Available: O'Reilly Online Learning (formerly Safari Books Online).
- K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
- Z. Li, Q. Ding, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: classification algorithms,” in Cyber Threat Intelligence, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, 2018, pp. 71–92.

Key References

Datasets:

- Intrusion Detection Evaluation dataset (CIC-IDS2017). [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>. Accessed: May 28, 2021.
- A Realistic Cyber Defense dataset (CSE-CIC-IDS2018). [Online]. Available: <https://registry.opendata.aws/cse-cic-ids2018/>. Accessed: May 28, 2021.
- DDoS Evaluation Dataset (CICDDoS2019). [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html>. Accessed: May 28, 2021.
- RIPE NCC: RIPE Network Coordination Center. [Online]. Available: <http://www.ripe.net/data-tools/stats/ris/ris-raw-data> [May 2021].
- University of Oregon Route Views project. [Online]. Available: <http://www.routeviews.org> [May 2021].

Key References

Echo State Networks:

- H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” German Nat. Res. Center for Inf. Technol. GMD, Bonn, Germany, Tech. Rep. 148, 2001.
- M. Lukoševičius, H. Jaeger, and B. Schrauwen, “Reservoir Computing Trends”, KI. Künstliche Intelligenz (Oldenbourg), vol. 26, no. 4, pp. 365–371, Nov. 2012.
- M. Lukosevicius, “A practical guide to applying Echo State Networks,” in Neural Networks: Tricks of the Trade (2nd ed.), G. Montavon, G. B. Orr, and K. -R. Müller, Eds., Berlin, Heidelberg, Springer, 2012, vol. 7700, pp. 659–686.

Key References

- K. Bekshentayeva, M. Canute, Y.-M. Kim, D. Lee, A. Wong, “Network Intrusion Detection Using Various Deep Learning Approaches”, BC Artificial Intelligence Showcase, Vancouver, BC, Dec. 2019.
- L. Gonzalez Rios, Z. Li, K. Bekshentayeva, and Lj. Trajkovic, "Detection of denial of service attacks in communication networks," in Proc. IEEE Int. Symp. Circuits and Systems, Seville, Spain, Oct. 2020 (virtual).
- L. Gonzalez Rios, K. Bekshentayeva, M. Singh, S. Haeri, and Lj. Trajkovic, "Virtual network embedding for switch-centric data center networks," in Proc. IEEE Int. Symp. Circuits and Systems, Daegu, Korea, May 2021 (virtual).
- K. Bekshentayeva and Lj. Trajkovic, “Detection of Denial of Service Attacks using Echo State Networks,” in Proc. IEEE International Conference on Systems, Man, and Cybernetics, Melbourne, Australia, submitted.

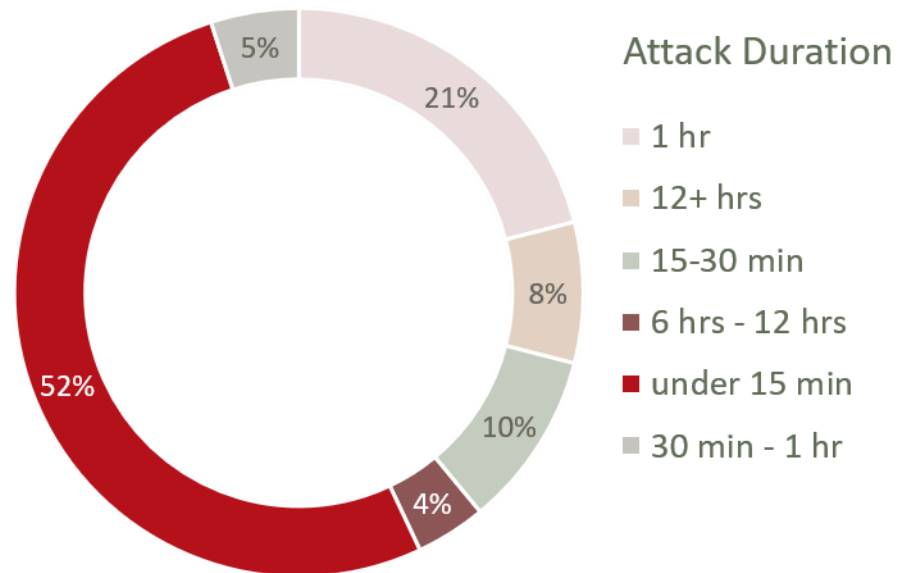


Thank you for your attention!

Questions

Denial of Service and Distributed Denial of Service (DoS and DDoS): Overview

- DoS and DDoS overload network's infrastructure causing disruptions and outages to small, medium, and large companies.



Types of DoS and DDoS attacks

Volumetric application/network level

flood a victim with voluminous requests, that may not be properly formatted, by consuming its bandwidth with UDP or ICMP packets until victim fails.

Amplification and reflection

may utilize connectionless nature of UDP. Requests are directed to the server with spoofed victim's address, and the amplified responses are reflected to the victim.

Network protocol attacks

usually target firewalls, load balancers, and servers by exploiting vulnerabilities of transport layer protocols.

Application protocol attacks

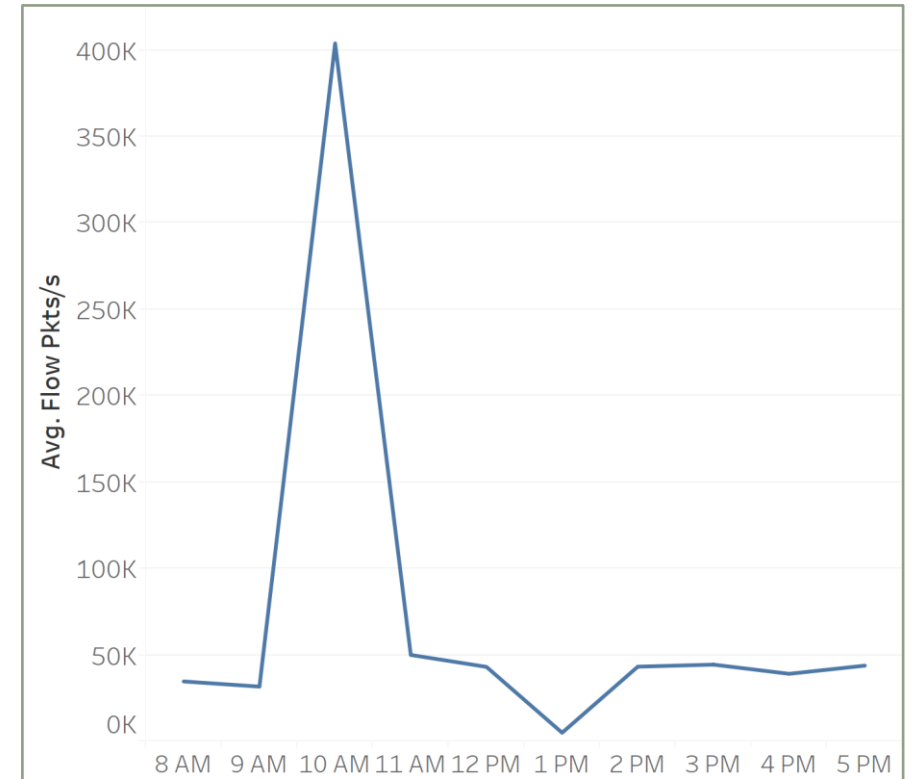
monopolize SMTP, HTTP, and DNS services.

Multivector attacks

combine various types of DoS/DDoS attacks. They may be launched as a flood and turn into other type of attack.

DoS/DDoS detection methods

- Due to the development of new attacks, a large number of **DoS/DDoS detection, mitigation, and prevention techniques** have been designed.
- The first step in countering an attack is **detection**, identifying that the attack is taking place. Older single source attacks, or volumetric attacks, by their nature, are detected easily by the majority of defense systems.
- DoS and DDoS detection methods include poll-based monitoring and detection, flow-based network parameter detection, network mirrors and deep packet inspection, and **anomalies-based detection**.

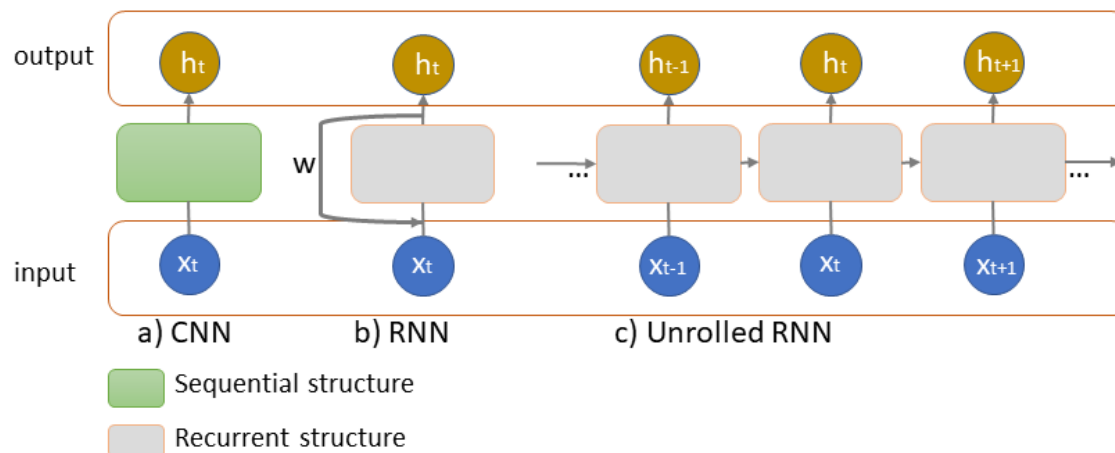


Overview of Machine Learning Algorithms used for Network Anomaly Detection

- Various **network anomaly detection systems** have been proposed that employ **machine learning algorithms** such as convolutional neural networks, recurrent neural networks (RNNs), deep belief networks, and autoencoders that offer promising results for anomaly detection.
- Support Vector Machine (SVM), Recurrent Neural Networks (LSTM, GRU), Broad Learning System (BLS) have been applied for data classification and intrusion detection in network traffic.
- Online anomaly detection framework that employs echo state network algorithm showed comparable accuracy to PC-based intrusion detection implementation.
 - ESN was a fast and simple approach that was not too resource intensive to be implemented on motes for pattern recognition.
 - ESN was proofed to detect a wider variety of anomalies with lower false alarm rate when compared to rule-based anomaly detection techniques.

Recurrent Neural Networks (RNN)

- RNNs belong to a class of artificial neural networks. They are widely used to detect anomalies in time-series datasets.
 - At each step the input of RNN is coming from a previous state of a hidden layer and is fed into the next state along with the new input signal. Often employ back propagation algorithm for training.



- Treat vanishing and exploding gradients (encountered in BPTT of the simplest RNN models) by efficiently coping with long-term dependencies: e.g., long short-term memory (LSTM). Reservoir Computing (RC) algorithms do not employ gradient-based optimization methods, thus, they don't encounter vanishing/exploding gradients.

ESN: Training

- For classification tasks a model is trained to decide a class for an input sequence given a $y^{target}(n)$ that is equal to 1 for a class of interest and 0 otherwise
- The class is decided by:

$$class\ x(n) = \operatorname{argmax}_k \left(\frac{1}{|\tau|} \sum_{n \in \tau} y_k(n) \right) = \operatorname{argmax}_k \left(\left(\sum y \right)_k \right)$$

Where τ is integration interval, $\sum y$ is time-averaged over τ

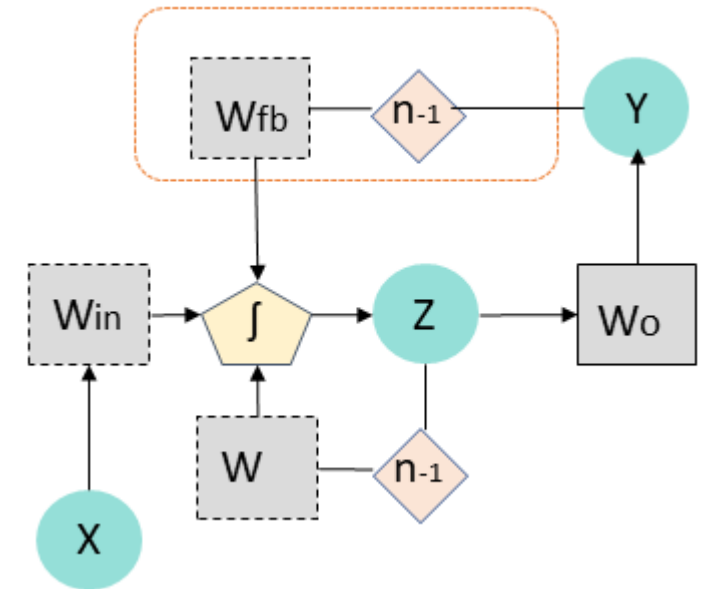
Creating ESN Reservoir

Input weights W_{in}

- Input weights W_{in} (initialized to None (no value)) depend on input size and are adjusted when the input is provided
- Binomial distribution, with $n = 1$, $p = 0.5$
- Size = $N_x \times N_z$

Reservoir weights W

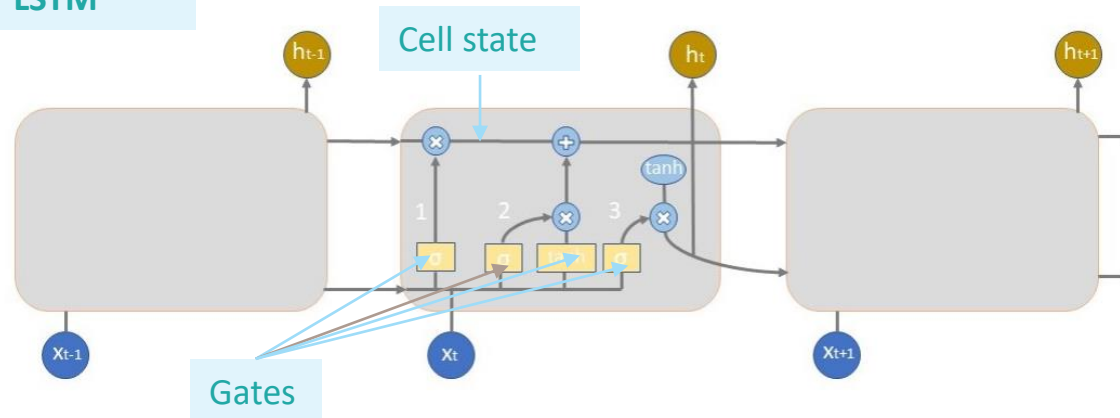
- Sparse connections yield better performance and speed up the updates: connectivity 25%
- Uniformly distributed between $[-0.5, 0.5]$ centered around zero
- Size = $N_z \times N_z$



- May be ambiguous what hyperparameters of the reservoir are responsible for which ESN's strengths or weaknesses when working on a particular task. Therefore, good and thorough hyperparameters initialization approach may improve ESN's predictive or classification capabilities.

Performance Results: ESN and Bi-LSTM

LSTM



1. Forget gate layer decides what information may be kept or discarded by accepting its inputs (output of the previous hidden state $h_{(t-1)}$ and a new input $x(t)$) and applying sigmoid activation.

$$f_t = \sigma(W^f(h_{t-1}, x_t) + b_f)$$

2. Input gate layer determines what new information is added:

$$i_t = \sigma(W^i(h_{t-1}, x_t) + b_i)$$

A vector \hat{C}_t of new candidate values to be added to the current state is created after tanh is applied:

$$\hat{C}_t = \tanh(W^c(h_{t-1}, x_t) + b_c)$$

Then, the current cell state C_t is updated as:

$$C_t = f_t \times C(t-1) + i_t \times \hat{C}_t$$

3. The sigmoid output gate layer o_t has a returning value of:

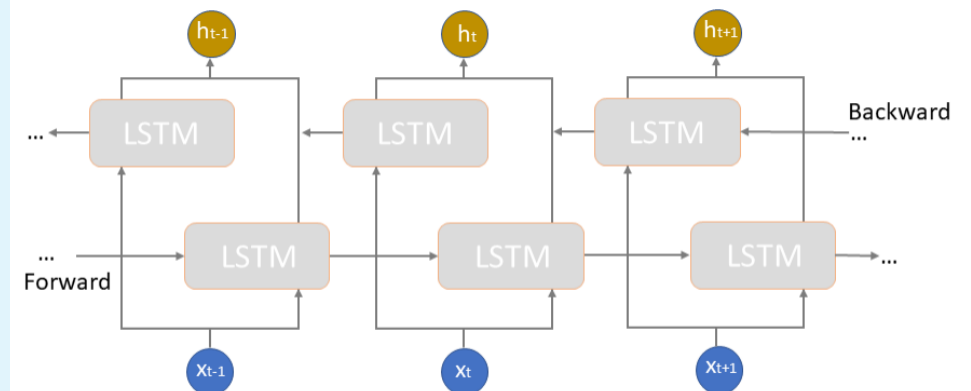
$$o_t = \sigma(W^o(h_{t-1}, x_t) + b_o)$$

that is multiplied by a vector of all possible values between -1 and 1 generated after applying tanh:

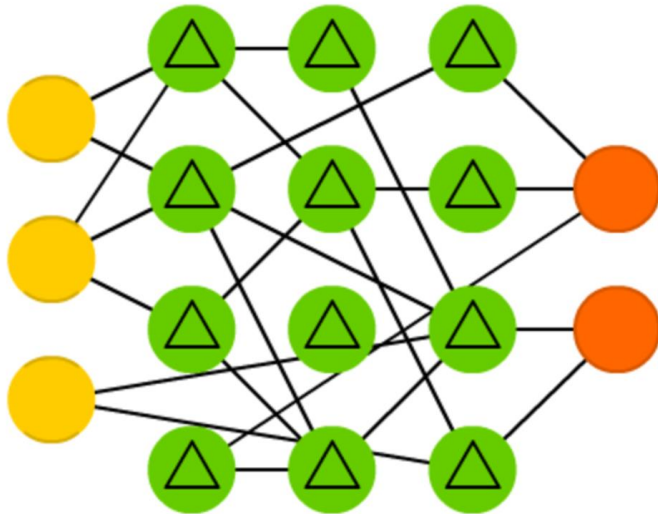
$$h_t = o_t \times \tanh C(t)$$

Bi-LSTM

- Variant of LSTM that has two hidden layers of opposite directions connected to the same output
- Improve the performance of a model for sequence classification tasks due to use of additional information, i.e. the output obtains information from past (backward, or negative time direction) and future (forward, or positive time direction) states at the same time
- Employ two LSTMs instead of one on the input sequence



Liquid State Machine (LSM)

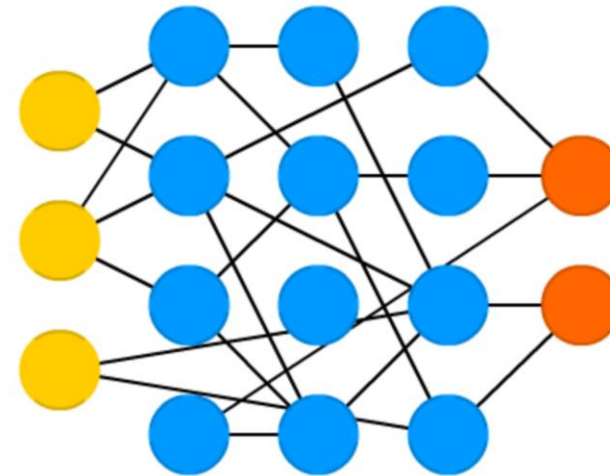


but without major breakthroughs.

LSM is sparse (not fully connected) neural network where activation functions are replaced by threshold levels. Cell accumulates values from sequential samples, and emits output only when the threshold is reached, setting internal counter again to zero.

Such idea is taken from human brain, and these networks are widely used in computer vision and speech recognition systems,

Echo State Network (ESN)









ESN is a subtype of recurrent networks with a special training approach. The data is passed to input, then the output if being monitored for multiple iterations (allowing the recurrent features to kick in). Only weights between hidden cells are updated after that.

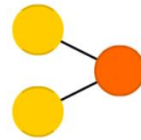
Personally, I know no real application of that type apart of multiple theoretical benchmarks. Feel free to add yours).

Neural Networks

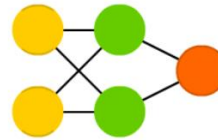
©2016 Fjodor van Veen - asimovinstitute.org

-  Backfed Input Cell
-  Input Cell
-  Noisy Input Cell
-  Hidden Cell
-  Probabilistic Hidden Cell
-  Spiking Hidden Cell
-  Output Cell
-  Match Input Output Cell
-  Recurrent Cell
-  Memory Cell
-  Different Memory Cell
-  Kernel
-  Convolution or Pool

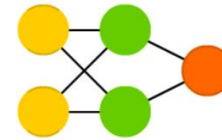
Perceptron (P)



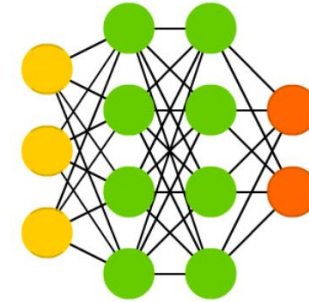
Feed Forward (FF)



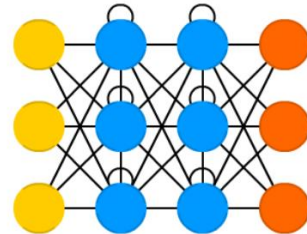
Radial Basis Network (RBF)



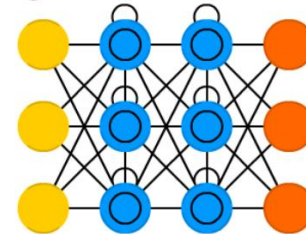
Deep Feed Forward (DFF)



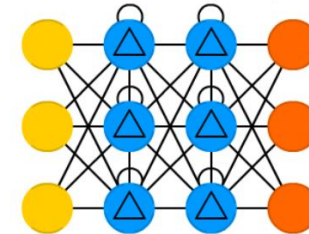
Recurrent Neural Network (RNN)



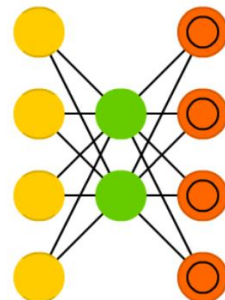
Long / Short Term Memory (LSTM)



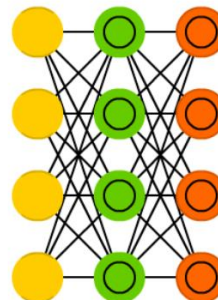
Gated Recurrent Unit (GRU)



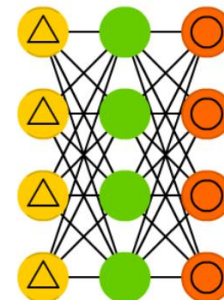
Auto Encoder (AE)



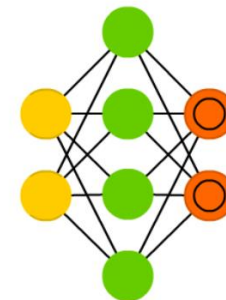
Variational AE (VAE)



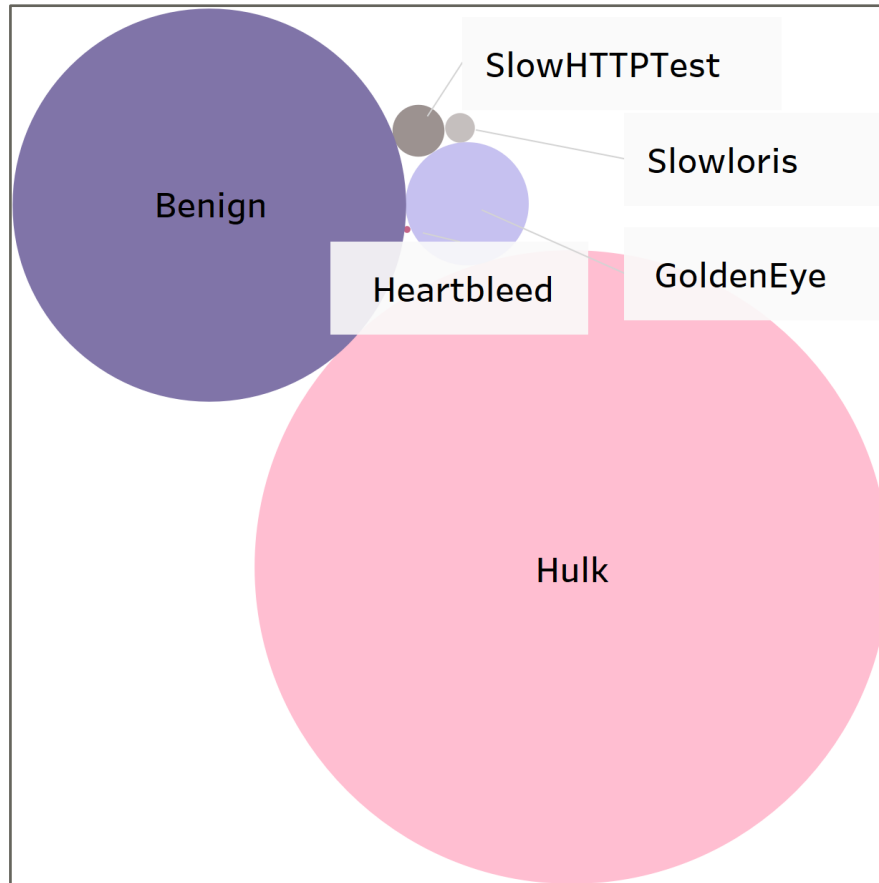
Denoising AE (DAE)



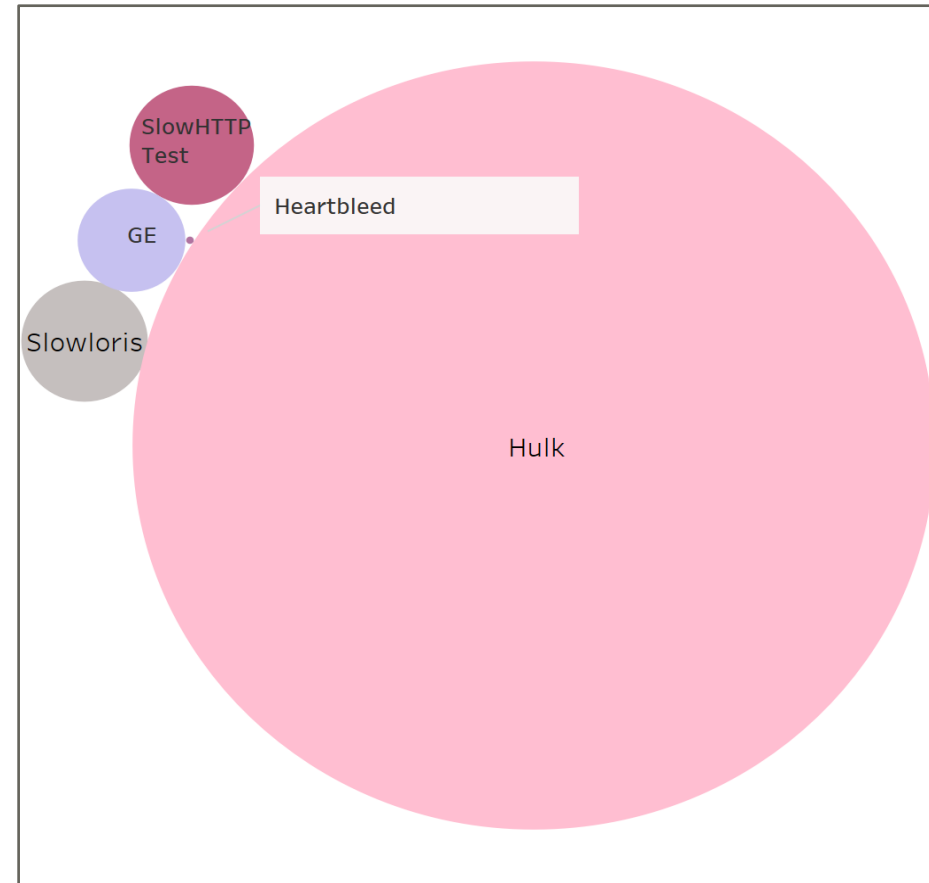
Sparse AE (SAE)



CIC-IDS2017

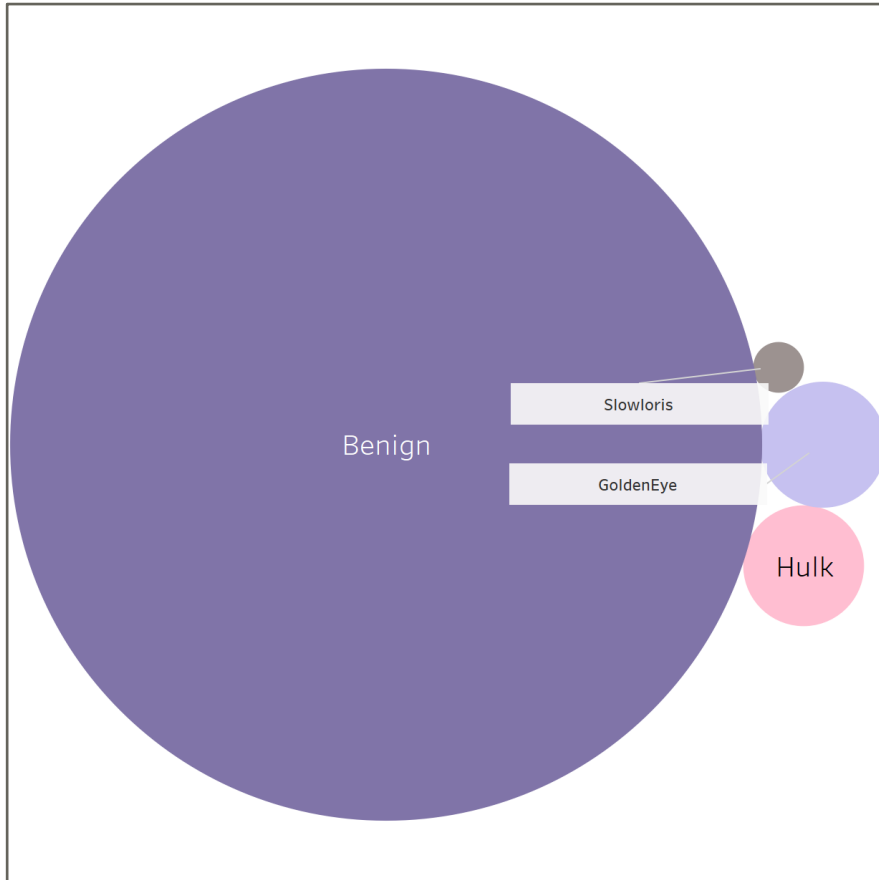


Average packet size

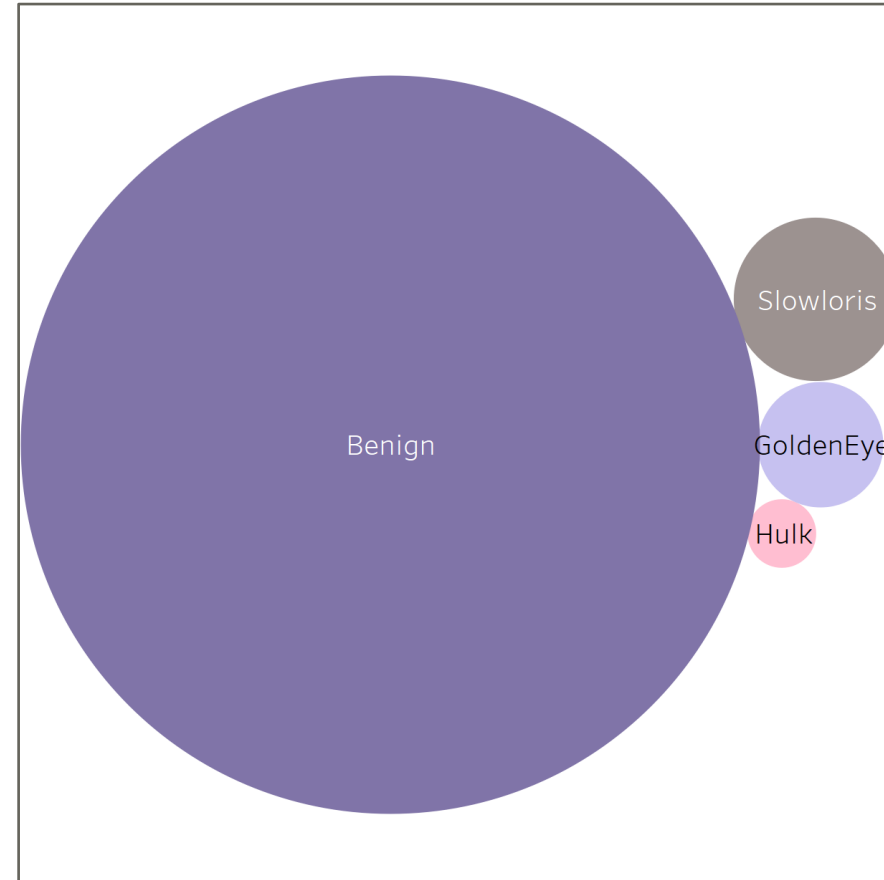


Average flow duration

CSE-CIC-IDS2018

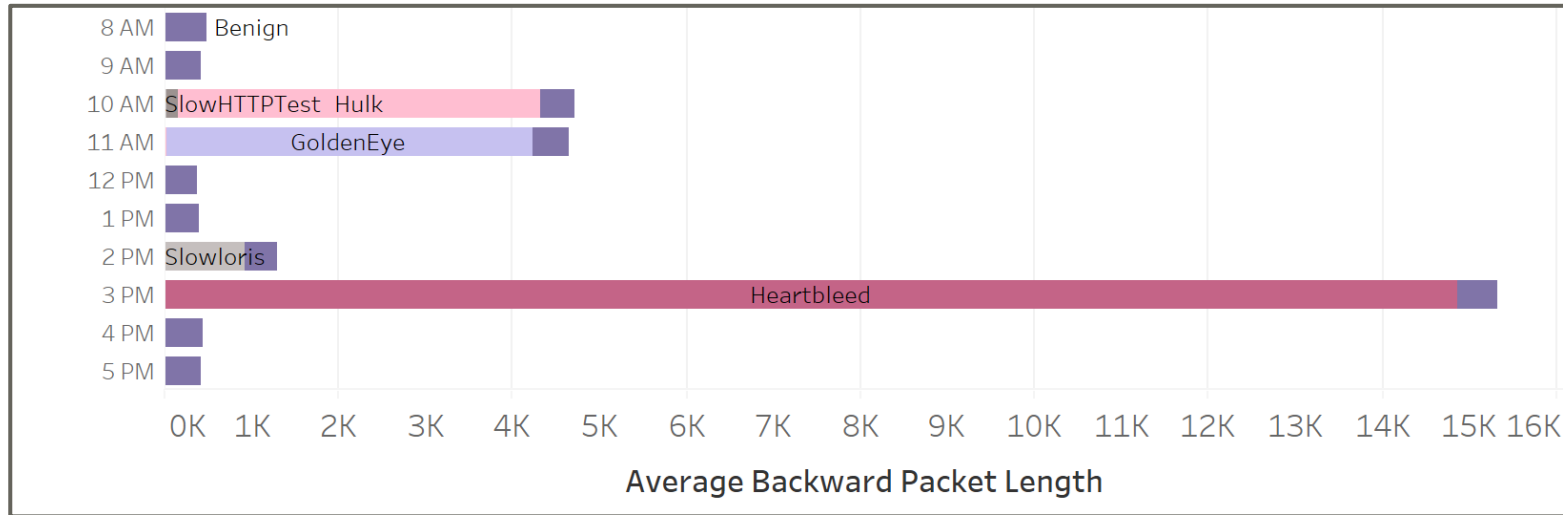


Average packet size



Average flow duration

Features



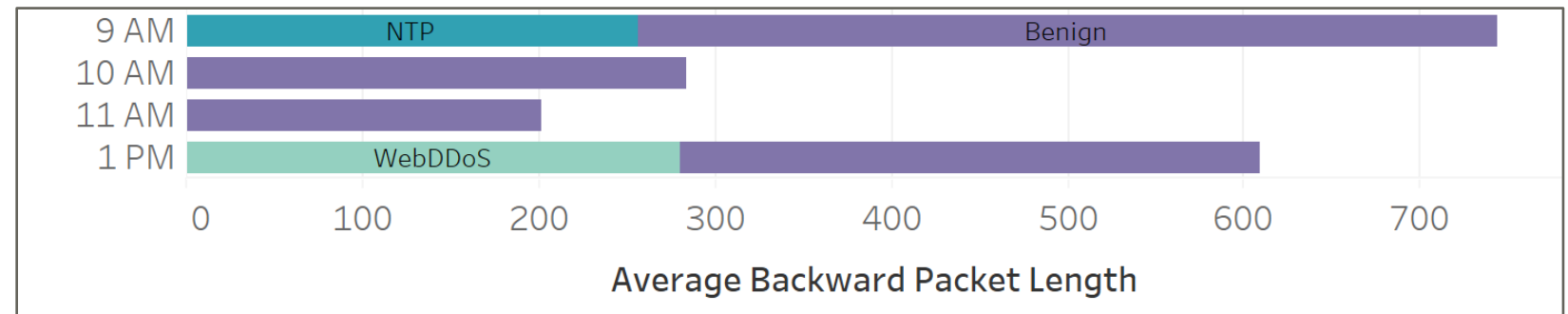
Packet length (CIC-IDS2017):

- Benign packets are generally under 1,000 bytes
- Heartbleed attack packets approximately reach 15,000 bytes on average.

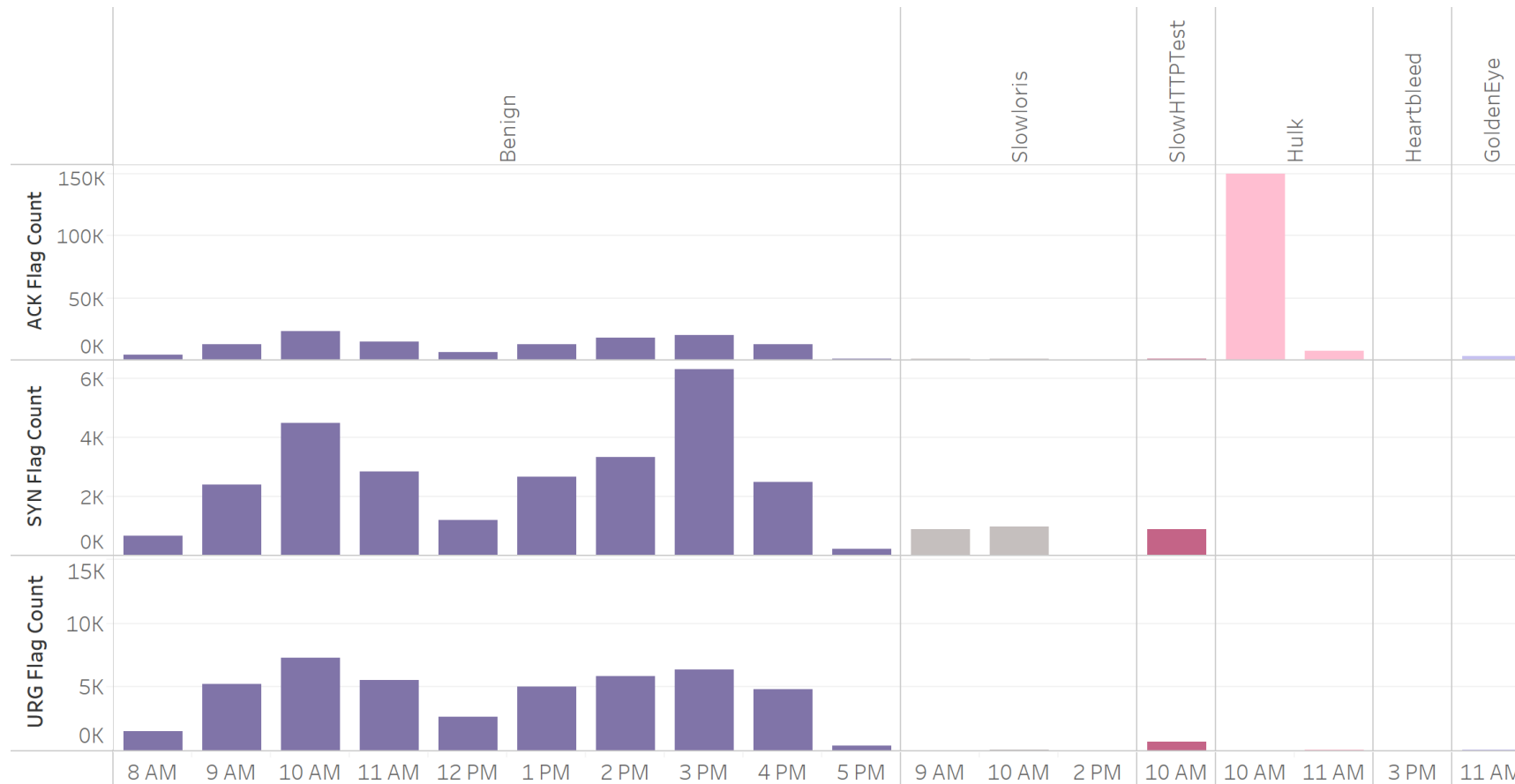
Packet length

(CIC-DDoS2019):

- The length of NTP and WebDDoS packets is smaller or comparable to benign



Features



TCP Flags (CIC-IDS2017):

- Hulk attack employs a large amount of packets with ACK set to 1

CIC-IDS2017 and CSE-CIC-IDS2018 Attacks

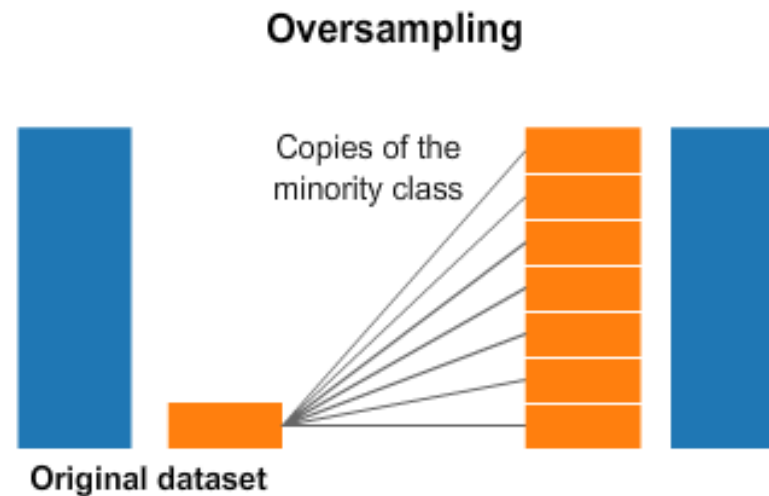
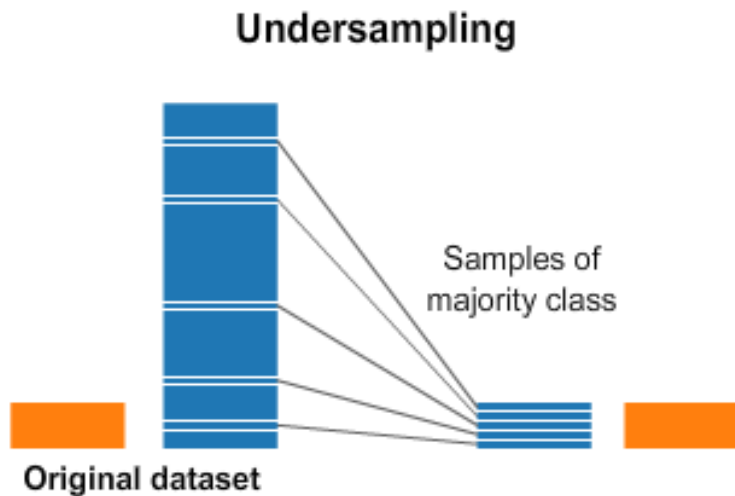
Attack	Description
GoldenEye	Application protocol attack; sends <i>keep alive</i> requests consuming available HTTP/HTTPS sockets
HULK	Volumetric application/network flood attack, brings down the servers with voluminous requests
SlowHTTPTest	Application protocol attack, similar to GoldenEye
Slowloris	Application protocol attack; sends fractional HTTP <i>get</i> requests without termination code

CIC-DDoS2019 Attacks

Attack	Description
Lightweight Directory Access Protocol (LDAP)	Utilizes LDAP via sending requests to a publicly available vulnerable LDAP server with open TCP port 389, which triggers (approximately 50 times larger than initial small queries) amplified replies reflected to a target server.
Network Time Protocol (NTP)	NTP amplification attack allows an attacker to use spoofed IP address of the victim's NTP infrastructure and send small NTP queries to the Internet servers that, in turn, generates and reflects amplified NTP responses.
SYN flood	Created via distributed botnet, overwhelms available resources of the target systems; may affect firewalls or other defense components of a target. SYN packets are sent to victim at a very high rate causing legitimate packets drop or element reboot. Approximately 80% of all DDoS attacks in 2018 were SYN floods
UDP-lag	Disrupts the connection between the client and the server. An example is in online gaming where the players wish to slow down/interrupt their opponents. Initiated either via a hardware switch known as a lag switch, or by a software program that allows monopolization of bandwidth
Web DoS	Same/different pages are requested constantly/N times per time period

Data Preprocessing: CIC-IDS2017, CSE-CIC-IDS2018, and CICDDoS2019 Resampling

- Oversampling and undersampling are the **resampling** techniques applied to imbalanced datasets with skewed class distribution
- **Random Oversampling** is randomly selecting samples from the minority class, and including them with replacement to dataset achieving the desired split
- **Undersampling** is randomly selecting samples from the majority class to remove from the dataset



Border Gateway Protocol Data Collections

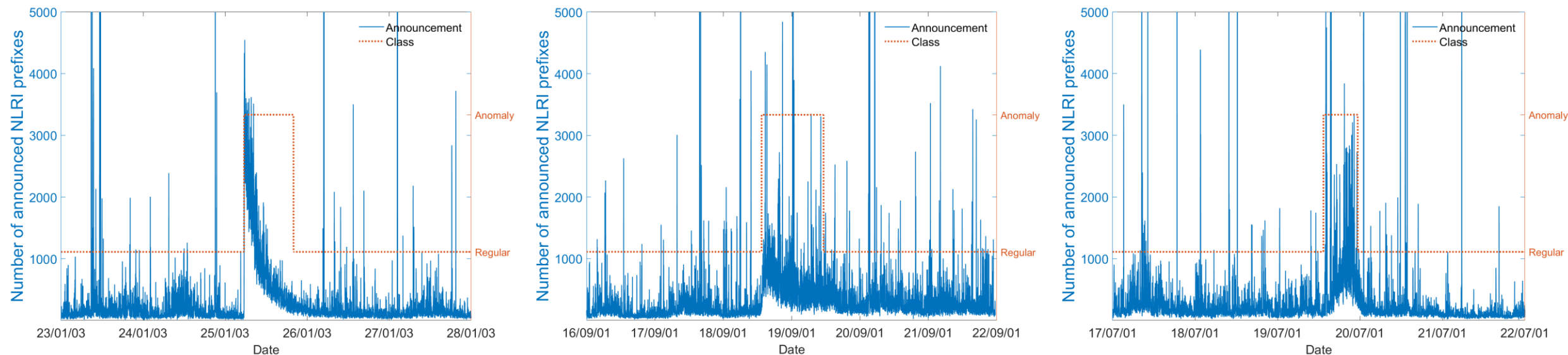
- BGP routing update messages are available from global BGP monitoring systems such as RIPE and Route Views.
- **RIS project** was launched in 2001 by the Réseaux IP Européens (**RIPE**) Network Coordination Centre (NCC) with the **main goal** to collect and store chronological routing data that offer a unique view of the Internet topology.
- The Internet routing data that contains BGP anomalous events: **Slammer, Nimda, and Code Red I** as well as **AWS (Amazon Web Services) DDoS attacks** used in this Thesis were acquired from RIPE (RIS project): **rrc04** (Geneva) and **rrc14** (Palo Alto) and **Route Views**

Border Gateway Protocol Datasets

Event	Beginning	Duration (min)
Slammer	25.01.2003	869
Nimda	18.09.2001	1301
Code Red I	19.07.2001	600

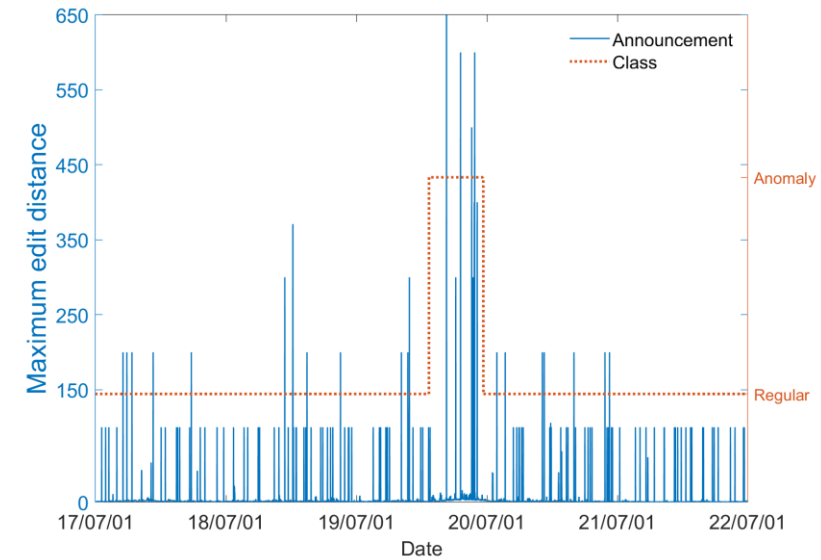
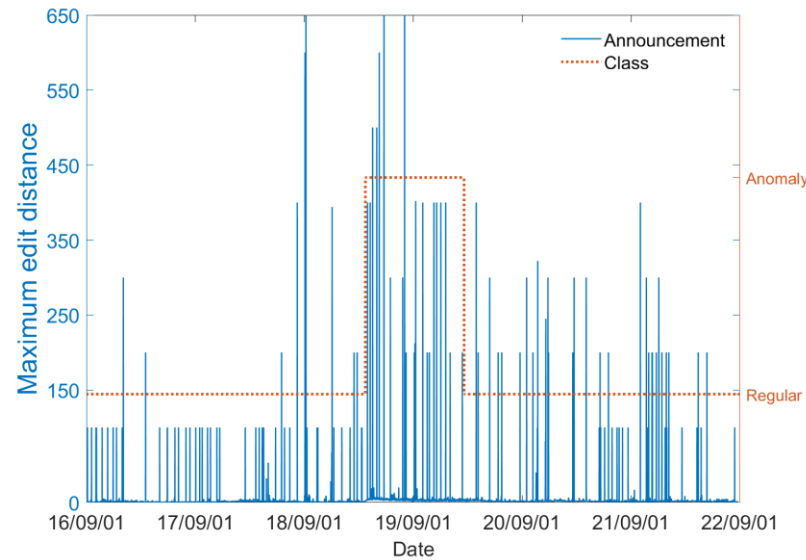
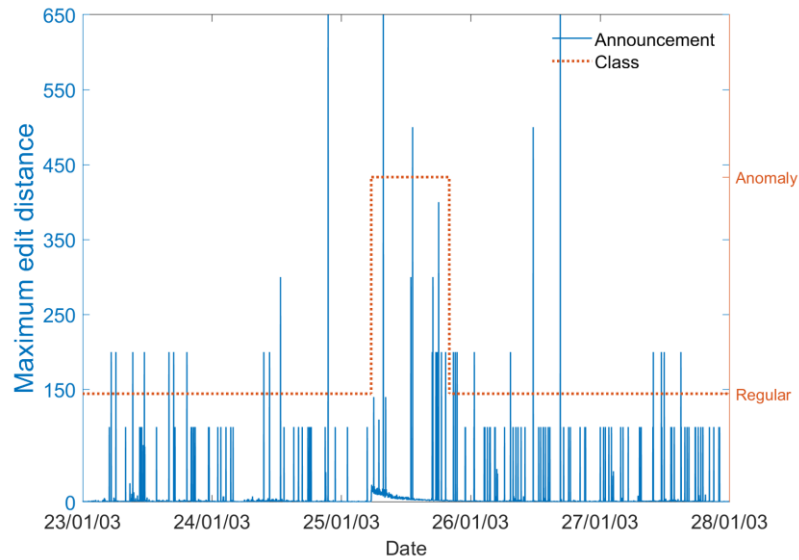
- **Slammer:** infected Microsoft SQL servers through a small piece of code. Furthermore, code replicated itself by infecting new machines through randomly generated targets (number of infected machines doubled approximately every nine seconds).
- **Nimda:** exploited vulnerabilities in the Microsoft Internet Information Services (IIS) web servers, propagated fast through email messages, web browsers, and file systems.
- **Code Red I:** affected approximately half a million IP addresses a day. Searched for vulnerable servers to infect and triggered buffer overflow.

RIPE: Slammer, Nimda, Code Red I - Feature 3



Number of announced NLRI prefixes
Slammer (left), Nimda (center), and Code Red I (right).
The red dotted line indicates the class.

RIPE: Slammer, Nimda, Code Red I - Feature 12

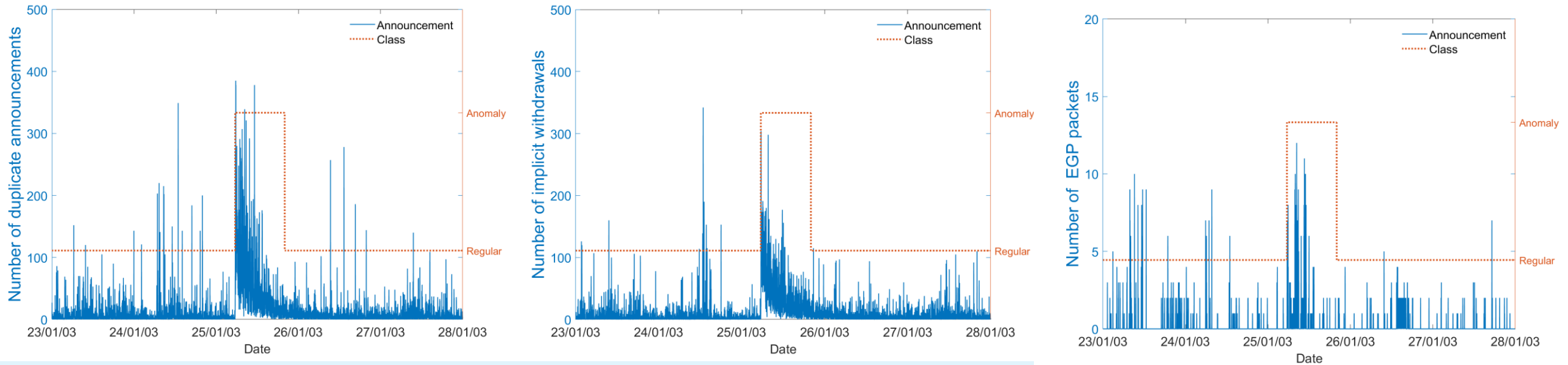


Maximum edit distance

Slammer (left), Nimda (center), and Code Red I (right).

The edit distance is a metric to quantify the similarity of strings. A router uses edit distance to measure the difference between two AS paths. The edit distance between two AS-path attributes is the minimum number of deletions, insertions, or substitutions that need to be executed to match the two attributes.

RIPE: Slammer - Feature 8, 10, 35



Slammer: Number of duplicate announcements (left), number of implicit withdrawals (center), and number of EGP packets (right)

- Duplicate announcements are the BGP update packets that have identical NLRI prefixes and the AS-path attributes.
- Implicit withdrawals are prefixes implicitly withdrawn by sending the same prefix with new attributes.
- Under a worm attack, BGP traces contained large volume of EGP packets.

Feature Selection

CIC-IDS2017 Wednesday, July 5

- | | |
|---|---|
| 1. feature 84: BiFlowsCount (0.189) | 11. feature 58: Average Packet Size (0.026) |
| 2. feature 5: Protocol (0.049) | 12. feature 46: Packet Length Mean (0.025) |
| 3. feature 53: ACK Flag Count (0.044) | 13. feature 29: Fwd IAT Max (0.025) |
| 4. feature 80: Idle Mean (0.039) | 14. feature 2: Source Port (0.024) |
| 5. feature 18: Bwd Packet Length Mean (0.038) | 15. feature 75: min_seg_size_forward (0.023) |
| 6. feature 24: Flow IAT Max (0.036) | 16. feature 19: Bwd Packet Length Std (0.021) |
| 7. feature 4: Destination Port (0.034) | 17. feature 44: Min Packet Length (0.020) |
| 8. feature 60: Avg Bwd Segment Size (0.033) | 18. feature 54: URG Flag Count (0.017) |
| 9. feature 47: Packet Length Std (0.029) | 19. feature 16: Bwd Packet Length Max (0.017) |
| 10. feature 82: Idle Max (0.027) | 20. feature 28: Fwd IAT Std (0.017) |
-

CSE-CIC-IDS2018 Thursday, February 15

- | | |
|--|---|
| 1. feature 70: Fwd Seg Size Min (0.218) | 11. feature 68: Init Bwd Win Byts (0.020) |
| 2. feature 67: Init Fwd Win Byts (0.086) | 12. feature 18: Flow IAT Mean (0.020) |
| 3. feature 79: BiFlowsCount (0.069) | 13. feature 22: Fwd IAT Tot (0.020) |
| 4. feature 1: Protocol (0.043) | 14. feature 26: Fwd IAT Min (0.019) |
| 5. feature 0: Dst Port (0.041) | 15. feature 25: Fwd IAT Max (0.018) |
| 6. feature 48: PSH Flag Cnt (0.034) | 16. feature 23: Fwd IAT Mean (0.016) |
| 7. feature 12: Bwd Pkt Len Max (0.030) | 17. feature 75: Idle Mean (0.015) |
| 8. feature 15: Bwd Pkt Len Std (0.029) | 18. feature 3: Flow Duration (0.015) |
| 9. feature 49: ACK Flag Cnt (0.026) | 19. feature 20: Flow IAT Max (0.015) |
| 10. feature 21: Flow IAT Min (0.021) | 20. feature 41: Pkt Len Max (0.015) |
-

Feature Selection

CSE-CIC-IDS2018 Friday, February 16

- | | |
|---|---|
| 1. feature 0: Dst Port (0.246) | 11. feature 44: Pkt Len Var (0.024) |
| 2. feature 8: Fwd Pkt Len Max (0.128) | 12. feature 37: Bwd Header Len (0.023) |
| 3. feature 11: Fwd Pkt Len Std (0.095) | 13. feature 19: Flow IAT Std (0.018) |
| 4. feature 41: Pkt Len Max (0.056) | 14. feature 54: Pkt Size Avg (0.018) |
| 5. feature 55: Fwd Seg Size Avg (0.047) | 15. feature 25: Fwd IAT Max (0.017) |
| 6. feature 10: Fwd Pkt Len Mean (0.046) | 16. feature 14: Bwd Pkt Len Mean (0.016) |
| 7. feature 43: Pkt Len Std (0.044) | 17. feature 68: Init Bwd Win Byts (0.015) |
| 8. feature 15: Bwd Pkt Len Std (0.033) | 18. feature 48: PSH Flag Cnt (0.016) |
| 9. feature 12: Bwd Pkt Len Max (0.032) | 19. feature 67: Init Fwd Win Byts (0.012) |
| 10. feature 23: Fwd IAT Mean (0.026) | 20. feature 6: TotLen Fwd Pkts (0.011) |
-

CIC-DDoS2019 Saturday, January 12

- | | |
|--|--|
| 1. feature 85: BiFlowsCount (0.206) | 11. feature 21: Flow Packets/s (0.029) |
| 2. feature 2: Source Port (0.086) | 12. feature 46: Packet Length Mean (0.021) |
| 3. feature 4: Destination Port (0.081) | 13. feature 55: CWE Flag Count (0.020) |
| 4. feature 54: URG Flag Count (0.076) | 14. feature 5: Protocol (0.020) |
| 5. feature 53: ACK Flag Count (0.056) | 15. feature 58: Average Packet Size (0.020) |
| 6. feature 13: Fwd Packet Length Min (0.051) | 16. feature 57: Down/Up Ratio (0.019) |
| 7. feature 59: Avg Fwd Segment Size (0.042) | 17. feature 20: Flow Bytes/s (0.018) |
| 8. feature 42: Fwd Packets/s (0.038) | 18. feature 72: Init_Win_bytes_forward (0.014) |
| 9. feature 44: Min Packet Length (0.038) | 19. feature 51: RST Flag Count (0.011) |
| 10. feature 14: Fwd Packet Length Mean (0.030) | 20. feature 35: Bwd IAT Min (0.009) |
-

Feature Selection

Slammer

1. feature 34: IGP packets (0.116)	11. feature 37: Packet size (B) (0.029)
2. feature 1: Number of announcements (0.112)	12. feature 6: Maximum <i>AS-path</i> length (0.024)
3. feature 36: Number of incomplete packets (0.102)	13. feature 13: Interarrival time (0.023)
4. feature 3: Number of announced NLRI prefixes (0.094)	14. feature 7: Average unique <i>AS-path</i> length (0.020)
5. feature 9: Number of duplicate withdrawals (0.084)	15. feature 5: Average <i>AS-path</i> length (0.019)
6. feature 8: Number of duplicate announcements (0.073)	16. feature 11: Average edit distance (0.018)
7. feature 10: Number of implicit withdrawals (0.072)	17. feature 20: Maximum edit distance $n = 13$ (0.016)
8. feature 4: Number of withdrawn NLRI prefixes (0.071)	18. feature 35: Number of EGP packets (0.009)
9. feature 2: Number of withdrawals (0.043)	19. feature 28: Maximum <i>AS-path</i> length $n = 10$ (0.004)
10. feature 12: Maximum edit distance (0.031)	20. feature 26: Maximum <i>AS-path</i> length $n = 8$ (0.004)

Nimda

1. feature 34: Number of IGP packets (0.136)	11. feature 8: Number of duplicate announcements (0.047)
2. feature 1: Number of announcements (0.129)	12. feature 13: Interarrival time (0.023)
3. feature 3: Number of announced NLRI prefixes (0.100)	13. feature 7: Average unique <i>AS-path</i> length (0.020) (0.019)
4. feature 4: Number of withdrawn NLRI prefixes (0.079)	14. feature 5: Average <i>AS-path</i> length (0.019)
5. feature 9: Number of duplicate withdrawals (0.075)	15. feature 35: Number of EGP packets (0.013)
6. feature 12: Maximum edit distance (0.067)	16. feature 6: Maximum <i>AS-path</i> length (0.011)
7. feature 37: Packet size (B) (0.059)	17. feature 11: Average edit distance (0.010)
8. feature 2: Number of withdrawals (0.055)	18. feature 16: Maximum edit distance $n = 9$ (0.004)
9. feature 36: Number of incomplete packets (0.054)	19. feature 14: Maximum edit distance $n = 7$ (0.004)
10. feature 10: Number of implicit withdrawals (0.049)	20. feature 32: Maximum <i>AS-path</i> length $n = 14$ (0.004)

Code Red I

1. feature 34: Number of IGP packets (0.137)	11. feature 8: Number of duplicate announcements (0.049)
2. feature 1: Number of announcements (0.137)	12. feature 13: Interarrival time (0.025)
3. feature 3: Number of announced NLRI prefixes (0.096)	13. feature 7: Average unique <i>AS-path</i> length (0.021) (0.019)
4. feature 4: Number of withdrawn NLRI prefixes (0.079)	14. feature 5: Average <i>AS-path</i> length (0.020)
5. feature 9 : Number of duplicate withdrawals (0.070)	15. feature 35: Number of EGP packets (0.012)
6. feature 12: Maximum edit distance (0.065)	16. feature 11: Average edit distance (0.009)
7. feature 36: Number of incomplete packets (0.058)	17. feature 6: Maximum <i>AS-path</i> length (0.009)
8. feature 37: Packet size (B) (0.057)	18. feature 32: Maximum <i>AS-path</i> length $n = 14$ (0.004)
9. feature 2: Number of withdrawals (0.055)	19. feature 18: Maximum edit distance $n = 11$ (0.004)
10. feature 10: Number of implicit withdrawals (0.049)	20. feature 29: Maximum <i>AS-path</i> length $n = 11$ (0.003)

Feature Selection

AWS 2019

1. feature 35: EGP packets (0.116)	11. feature 12: Maximum edit distance (0.029)
2. feature 1: Number of announcements (0.112)	12. feature 4: Number of withdrawn NLRI prefixes (0.024)
3. feature 10: Number of implicit withdrawals (0.102)	13. feature 7: Average unique <i>AS-path</i> length (0.023)
4. feature 3: Number of announced NLRI prefixes (0.094)	14. feature 7: Average unique <i>AS-path</i> length (0.020)
5. feature 2: Number of withdrawals (0.084)	15. feature 13: Interarrival time (0.019)
6. feature 34: Number of IGP packets (0.073)	16. feature 11: Average edit distance (0.018)
7. feature 37: Packet size (B)(0.072)	17. feature 21: Maximum edit distance $n = 14$ (0.016)
8. feature 3: Number of announced NLRI prefixes (0.071)	18. feature 5: Average <i>AS-path</i> length (0.009)
9. feature 8: Number of duplicate announcements (0.043)	19. feature 20: Maximum edit distance $n = 13$ (0.004)
10. feature 9: Number of duplicate withdrawals (0.031)	20. feature 22: feature 22: Maximum edit distance $n = 15$ (0.004)

AWS 2020

1. feature 8: Number of duplicate announcements (0.136)	11. feature 12: Maximum edit distance (0.047)
2. feature 2: Number of withdrawals(0.129)	12. feature 35: Number of EGP packets (0.023)
3. feature 9: Number of duplicate withdrawals (0.100)	13. feature 13: Interarrival time (0.019)
4. feature 36: Number of incomplete packets (0.009)	14. feature 11: Average edit distance (0.019)
5. feature 3: Number of announced NLRI prefixes (0.075)	15. feature 6: Maximum <i>AS-path</i> length (0.013)
6. feature 34: Number of IGP packets (0.013) (0.004)	16. feature 5: Average <i>AS-path</i> length (0.011)
7. feature 1: Number of announcements (0.059)	17. feature 7: Average unique <i>AS-path</i> length (0.010)
8. feature 37: Packet size (B) (0.055)	18. feature 20: Maximum edit distance $n = 13$ (0.004)
9. feature 4: Number of withdrawn NLRI prefixes (0.054)	19. feature 23: Maximum edit distance $n = 16$ (0.004)
10. feature 10: Number of implicit withdrawals (0.049)	20. feature 22: Maximum edit distance $n = 15$ (0.004)

Libraries/packages

Experiments

Windows 10 64-bit Operating System and Intel Core i7-8650U CPU at 1.9-2.11 GHz;
Python 3.8

Numpy

supports large and multidimensional arrays and matrices, and high-level mathematical functions to manipulate these arrays and matrices

Pandas

data manipulation and analysis

Scipy

functions to work around with different format of files

- **"sparse"** for sparse matrices manipulation
- **scipy.io**: allows reading data from and write data to a variety of file formats
- **scipy.stats**: a module that includes a large number of probability distributions, along with a growing library of statistical functions

Scikit-learn/sklearn

contains various classification, regression, and clustering algorithms; designed to interoperate with numpy and scipy;

- **"preprocessing"**
- **sklearn.decomposition "PCA"** for linear dimensionality reduction
- **sklearn.linear_model "Ridge"** for linear least squares with L2 regularization
- **sklearn.metrics "accuracy_score", "f1_score, sklearn.model_selection "train_test_split"**

Math

provides an access to mathematical functions defined by the C standard

Performance Metrics

Confusion matrix conditions:

- True Negative (TN): the model correctly classifies regular data points as regular
- False Negative (FN): the model incorrectly classifies anomalous data points as regular
- False Positive (FP): the model incorrectly classifies regular data points as anomaly
- True Positive (TP): the model correctly classifies anomalous data points as anomaly

Actual Class	Predicted Class	
	Negative (Regular)	Positive (Anomaly)
Negative (Regular)	TN	FP
Positive (Anomaly)	FN	TP

Performance Metrics

- **Accuracy** reflects the proportion of the results predicted accurately.
 - May be a misleading measure for imbalanced datasets because it accepts equal cost for misclassification despite of the distribution of classes.
 - $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$
- **F-Score** considers the false predictions and may be described as a harmonic mean of the precision and recall (or sensitivity).
 - It measures the discriminating ability of the classifier to identify classified and misclassified anomalies.
 - $F-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

Performance Metrics

- **Precision** identifies true anomalies among all data points that are classified as anomalies.

- $\text{Precision} = \frac{TP}{TP+FP}$

- **Recall (Sensitivity)** measures the ability of the model to identify correctly predicted anomalies.

- $\text{Recall} = \frac{TP}{TP+FN}$

- **Specificity**, or true negative rate, measures the proportion of actual negatives that are correctly identified.

- $\text{Specificity} = \frac{TN}{TN+FP}$

- **False alarm rate** (FAR) is a common measure used for evaluating intrusion detection models. It is equal to $1 - \text{Specificity}$, and also:

- $\text{FAR} = \frac{FP}{TN+FP}$

Performance Results

Dataset	Class	Number of instances	Training set	Test set
CIC-IDS2017, Wednesday, July 5	Total	346,352	277,081	69,271
	Regular	219,984	175,855	44,129
	Anomaly	126,368	101,226	25,142
CSE-CIC-IDS2018, Thursday, February 15	Total	525,288	419,430	104,858
	Regular	497,973	398,349	99,624
	Anomaly	26,315	21,081	5,234
CSE-CIC-IDS2018, Friday, February 16	Total	525,288	419,430	104,858
	Regular	223,208	178,483	44,725
	Anomaly	301,080	240,947	60,133
CIC-DDoS2019, Saturday, January 12	Total	500,000	400,000	100,000
	Regular	249,977	200,016	49,961
	Anomaly	250,023	199,984	50,039

Number of data points in training and test sets in
CIC-IDS2017 Wednesday, July 5 (unbalanced), CIC-CSE-IDS2018 Thursday, February 15 (unbalanced),
CIC-CSE-IDS2018 Friday, February 16 (balanced), and CIC-DDoS2019 Saturday, January 12 (balanced)

Performance Results: Oversampling and Undersampling

Dataset	Class	After oversampling	Training set	Test set
CIC-IDS2017, Wednesday, July 5	Total	400,000	320,000	80,000
	Regular	200,376	160,415	40,039
	Anomaly	199,624	159,585	39,961
CSE-CIC-IDS2018, Thursday, February 15	Total	500,000	400,000	100,000
	Regular	250,060	200,403	50,343
	Anomaly	249,940	199,597	49,657
CIC-DDoS2019, Saturday, January 12	Total	500,000	400,000	100,000
	Regular	249,977	200,016	50,039
	Anomaly	250,023	199,984	49,961
Dataset	Class	After undersampling	Training set	Test set
CIC-IDS2017, Wednesday, July 5	Total	252,672	202,137	50,535
	Regular	126,388	100,918	25,470
	Anomaly	126,284	101,219	25,065
CSE-CIC-IDS2018, Thursday, February 15	Total	52,498	41,998	10,500
	Regular	26,280	21,012	5,294
	Anomaly	26,218	20,986	5,206

Number of data points in training and test sets after oversampling (top) and undersampling (bottom)

Performance Results

Dataset	Class	Entire dataset	Training set	Test set
Slammer	Total	7,200	5,760	1,440
	Regular	6,331	5,058	1,273
	Anomaly	869	702	167
Nimda	Total	8,609	6,887	1,722
	Regular	7,308	5,841	1,467
	Anomaly	1,301	1,046	255
Code Red I	Total	7,200	5,760	1,440
	Regular	6,600	5,272	1,328
	Anomaly	600	488	112

Number of data points in training and test sets in
BGP datasets: Slammer, Nimda, Code Red I

Performance Results

Dataset	Class	Entire dataset	Training set	Test set
DDoS2019	Total	10,080	6,048	4,032
	Regular	6,390	3,823	2,567
	Anomaly	3,690	2,225	1,465
DDoS2020	Total	10,080	8,064	2,016
	Regular	5,709	4,572	1,136
	Anomaly	4,371	3,492	880

Number of data points in training and test sets in
BGP datasets: DDoS2019_v1, DDoS2019_v2, and DDoS2020