

Data Mining and Machine Learning for Analysis of Network Traffic

Ljiljana Trajković

ljilja@cs.sfu.ca

Communication Networks Laboratory

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

School of Engineering Science

Simon Fraser University, Vancouver

British Columbia, Canada

Simon Fraser University Burnaby Campus



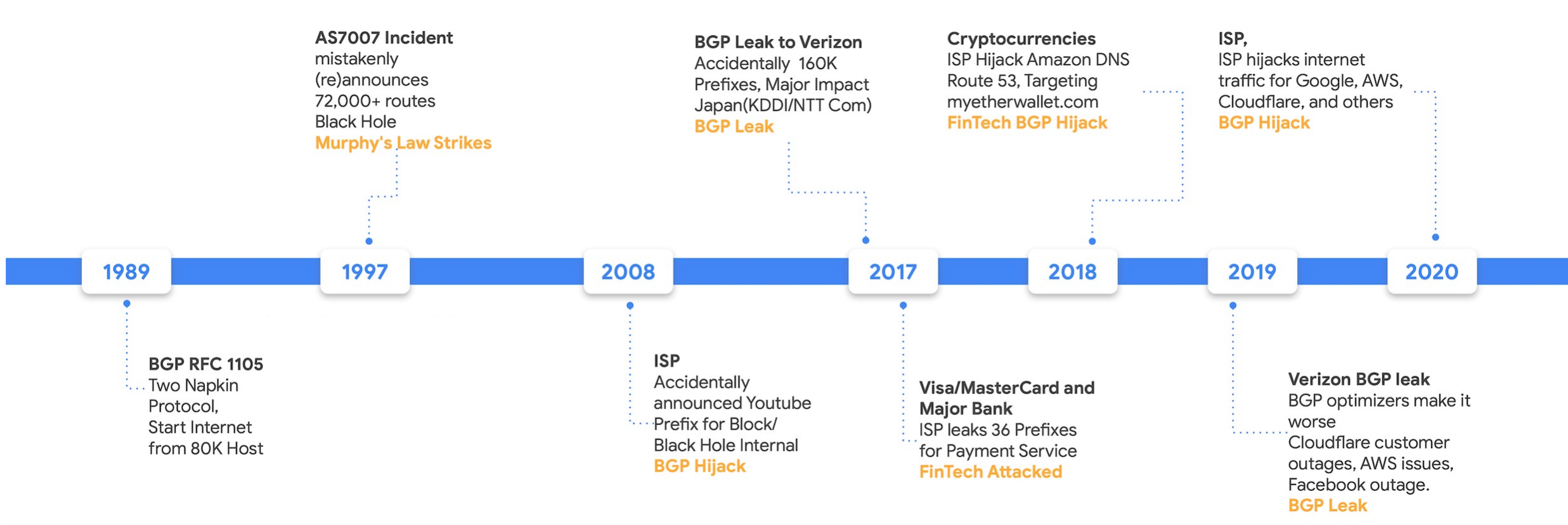
Roadmap

- Introduction
- Data collections
- Feature selection
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

Introduction

- The ubiquitous nature of the Internet renders it vulnerable to exploitation, particularly in geopolitical tensions or military conflicts
- Internet outages block communication channels, inhibit the flow of information, and affect adversaries' infrastructure
- Affected critical physical Internet infrastructure may include:
 - power grids, submarine fiber optic cables, terrestrial links, Internet exchange points
- The Internet remains susceptible to attacks despite new technological advances that enable using various types of connection media and mechanisms for redundancy

Brief History of Internet Anomalies: BGP Hijack/Leak Outages



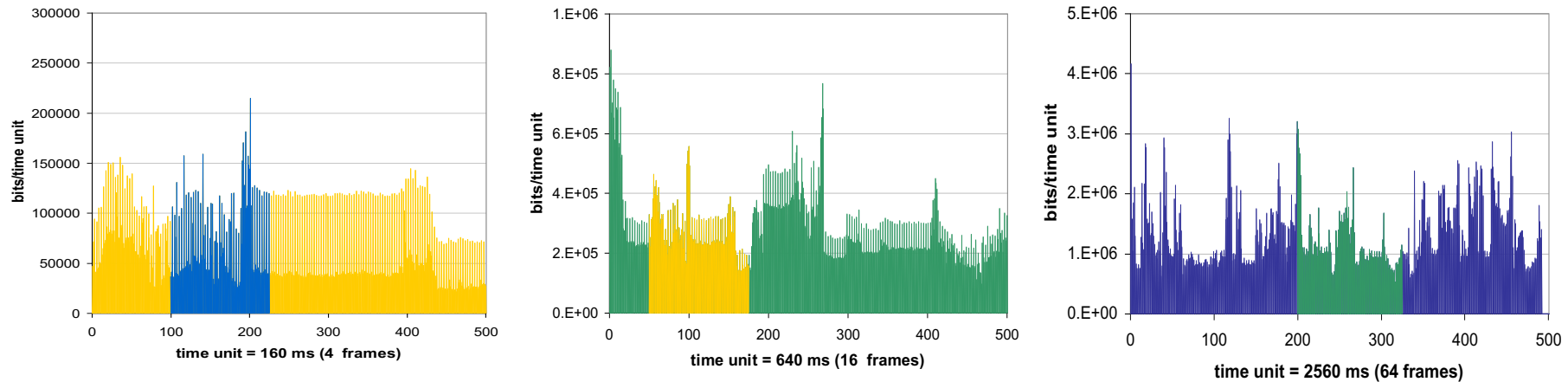
Source: <https://www.slideshare.net/mynog/cloud-sdn-bgp-peering-and-rpki>

Measurements of Network Traffic

- **Traffic measurements:**
 - basis for characteristics of network traffic and developing traffic models
 - used to evaluate performance of protocols and applications
- **Traffic analysis:**
 - provides information about the network usage
 - helps understand the behavior of network users
 - helps identify trends in the evolution of the Internet topology
 - **enables classification of traffic and detection of network anomalies**
- **Traffic prediction:**
 - important to assess future network capacity requirements
 - used to plan future network developments

Self-Similarity: Influence of Time-Scales

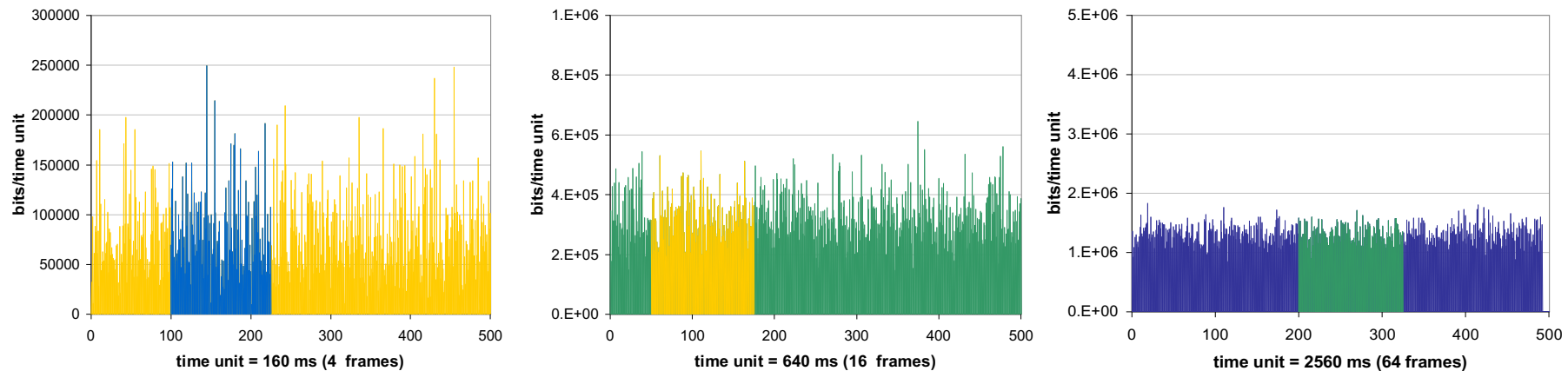
- Genuine MPEG traffic trace



W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” *IEEE/ACM Trans. Networking*, vol. 2, no 1, pp. 1-15, Feb. 1994.

Self-Similarity: Influence of Time-Scales

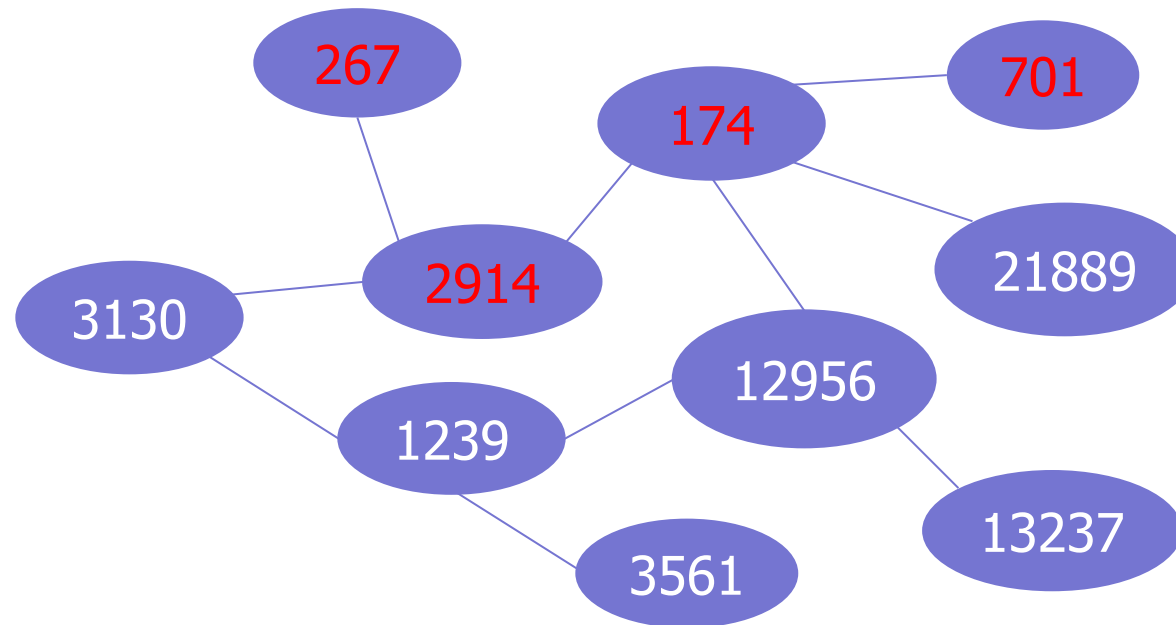
- Synthetically generated Poisson model



W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” IEEE/ACM Trans. Netw., vol. 2, no 1, pp. 1-15, Feb. 1994.

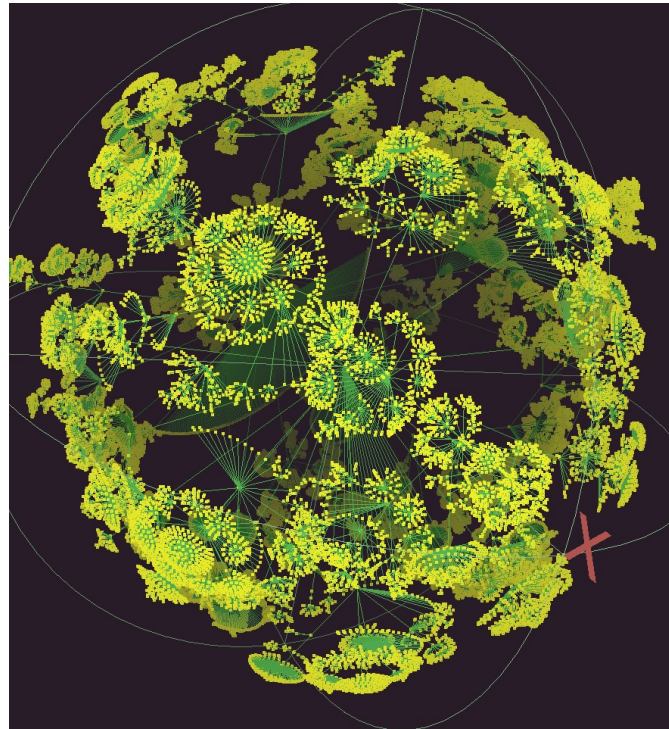
Internet Topology at AS Level

- Collected data from Border Gateway Protocols (BGP) routing tables are used to infer the Internet topology



G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, “Power-laws and the AS-level Internet topology,” IEEE/ACM Trans. Networking, vol. 11, no. 4, pp. 514–524, Aug. 2003.

The Internet Topology: Scale Free Graphs



<http://www.caida.org/home/>
1hr: 535,102 nodes and 601,678 links

G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, “Power-laws and the AS-level Internet topology,” *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 514–524, Aug. 2003.

Roadmap

- Introduction
- Data collections
- Feature selection
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

Network Traffic Datasets

- Internet Border Gateway Protocol (BGP) anomalies:
 - Computer worms and viruses:
 - Code Red (2001), Nimda (2001), Slammer (2003)
 - Electrical failures:
 - Moscow blackout (2005), Pakistan power outage (2021)
 - Ransomware attacks:
 - WannaCrypt (2017), WestRock (2021)
 - Regions of recent conflicts (2024): Iraq, The Gaza Strip, Ukraine

Network Traffic Datasets

- Border Gateway Protocol (BGP) routing data collected by:
 - Routing Information Service at Réseaux IP Européens (**RIPE**) Network Coordination Centre
 - **Route Views** project, University of Oregon
- Canadian Institute for Cybersecurity (**CIC**), University of New Brunswick
- Internet Outage Detection and Analysis (**IODA**) datasets:
 - Center for Applied Internet Data Analysis (CAIDA), University of California San Diego
 - Maintained by the Georgia Institute of Technology

BGP Datasets: Internet worms

- Slammer, Nimda, Code Red:

Collection site	Dataset	Regular (min)	Anomaly (min)	Regular (training)	Anomaly (training)	Regular (test)	Anomaly (test)	Start	End
RIPE	Code Red	6,600	600	3,679	361	2,921	239	17.07.2001 00:00:00	21.07.2001 23:59:59
	Nimda	7,308	1,301	3,673	827	3,635	474	16.09.2001 00:00:00	21.09.2001 23:59:59
	Slammer	6,331	869	3,210	530	3,121	339	23.01.2003 00:00:00	27.01.2003 23:59:59
Route Views	Slammer	6,319	869	3,198	530	3,121	339	23.01.2003 00:00:00	27.01.2003 23:59:59

Route Views data collection began in 2003.

BGP Datasets: Power Blackouts and Outages

Collection site	Dataset	Regular (min)	Anomaly (min)	Regular (training)	Anomaly (training)	Regular (test)	Anomaly (test)	Start	End
RIPE	Moscow blackout	6,960	240	3,120	180	3,840	60	23.05.2005 00:00:00	27.05.2005 23:59:59
	Pakistan power outage	6,880	320	4,000	200	2,880	120	07.01.2021 00:00:00	11.01.2021 23:59:59
Route Views	Moscow blackout	6,865	130	3,075	85	3,790	45	23.05.2005 00:00:00	27.05.2005 23:59:59
	Pakistan power outage	6,880	320	4,000	200	2,880	120	07.01.2021 00:00:00	11.01.2021 23:59:59

BGP Datasets: Ransomware Attacks

Collection site	Dataset	Regular (min)	Anomaly (min)	Regular (training)	Anomaly (training)	Regular (test)	Anomaly (test)	Start	End
RIPE/ Route Views	WannaCrypt	5,760	5,760	2,880	3,420	2,880	2,340	10.05.2017 00:00:00	17.05.2017 23:59:59
	WestRock ransomware	5,832	10,008	2,952	6,008	2,880	4,000	21.01.2021 00:00:00	31.01.2021 23:59:59

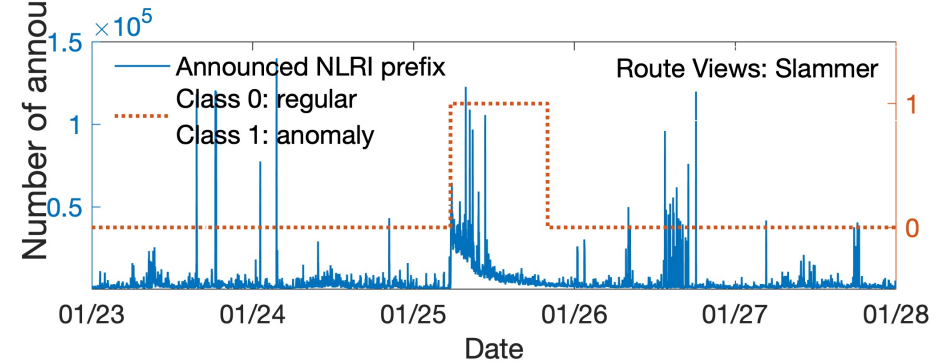
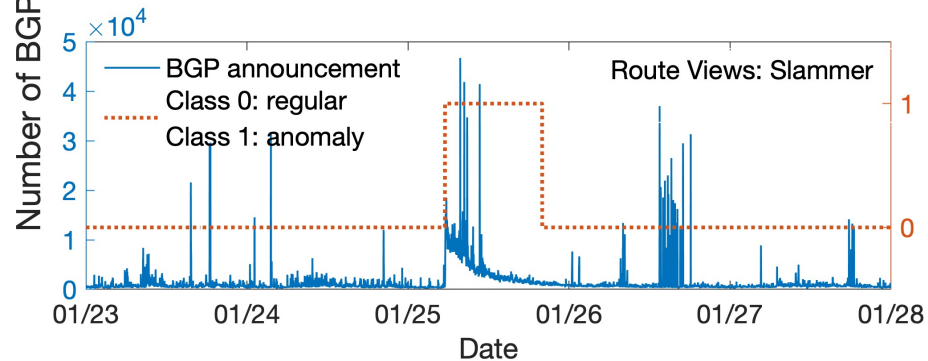
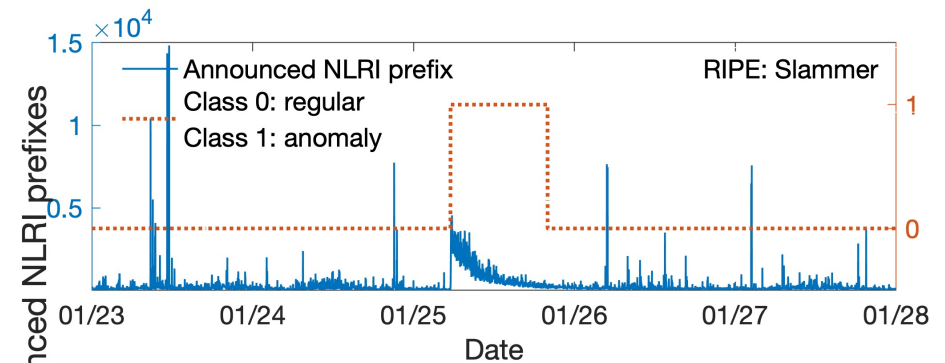
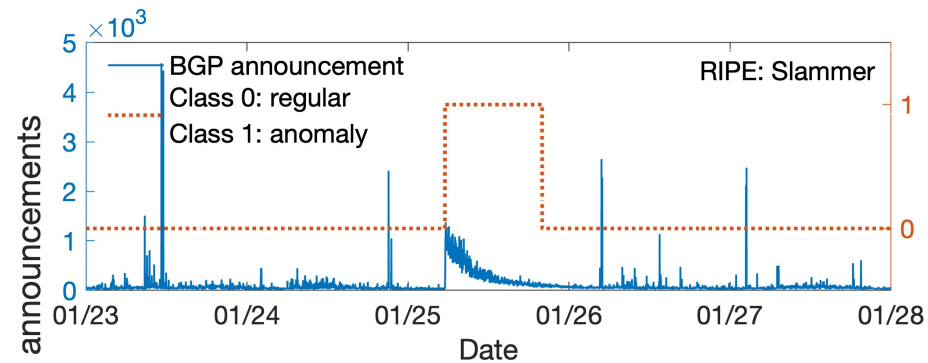
Outage Datasets

TABLE I
OUTAGES DATASETS: NUMBER OF DATA POINTS AND COLLECTION DATES

Collection Site	Dataset	Total		Training		Test		Collection Date	
		Regular	Anomaly	Regular	Anomaly	Regular	Anomaly	Begin	End
RIPE	Iraq	6,497	939	3,853	563	2,644	376	20/08/2023 01:00:00	25/08/2023 04:55:00
	Gaza Strip	13,323	3,468	5,734	2,080	7,589	1,388	26/10/2023 15:25:00	07/11/2023 07:15:00
	Ukraine	3,934	677	2,494	406	1,440	271	21/03/2022 09:30:00	24/03/2022 14:20:00
IODA	Iraq	2,545	383	1,537	229	1,008	154	20/08/2023 01:00:00	30/08/2023 04:55:00
	Gaza Strip	2,663	696	1,146	417	1,517	279	26/10/2023 15:25:00	07/11/2023 07:15:00
	Ukraine	786	137	498	82	288	55	21/03/2022 09:30:00	24/03/2022 14:20:00

BGP Dataset: Slammer (2003)

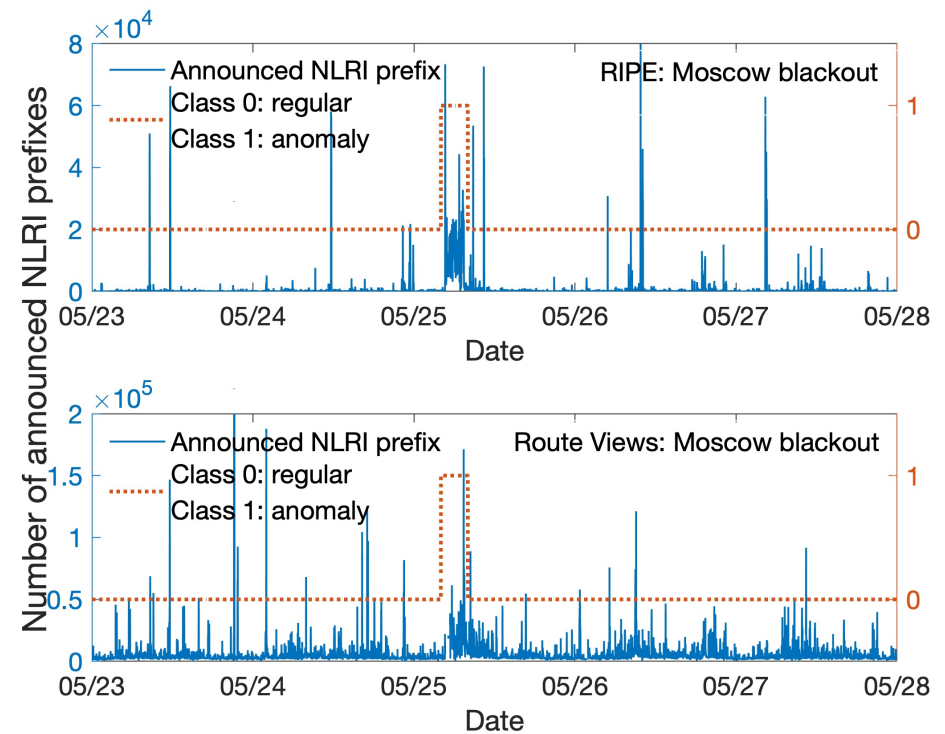
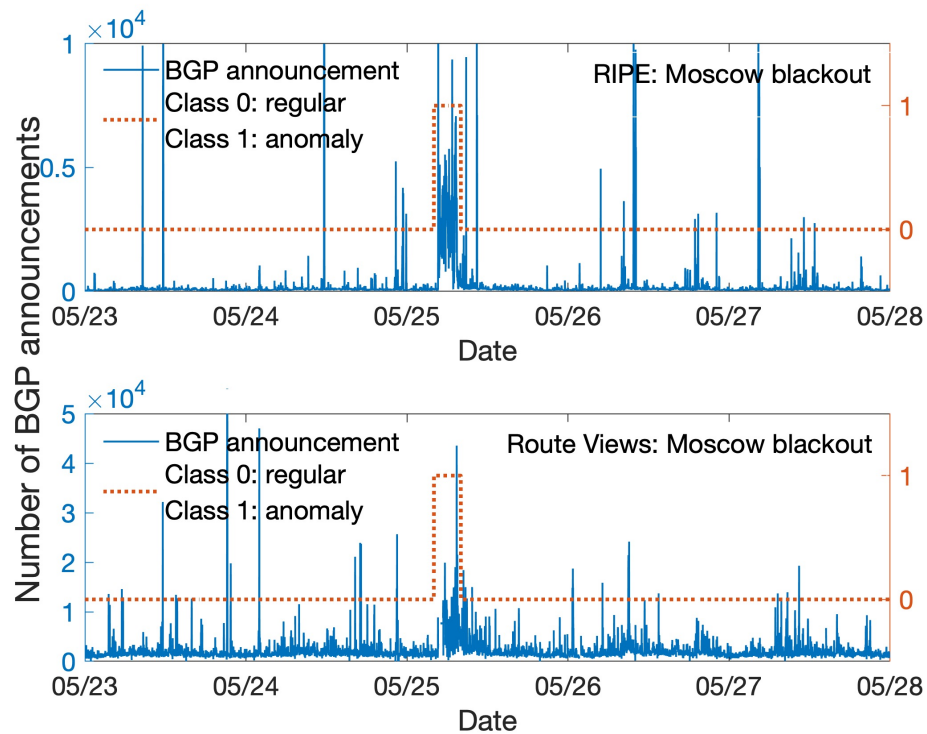
- Number of BGP announcements and announced NLRI prefixes vs. date:



BGP: Border Gateway Protocol
NLRI: Network Layer Reachability Information

BGP Dataset: Moscow Blackout (2005)

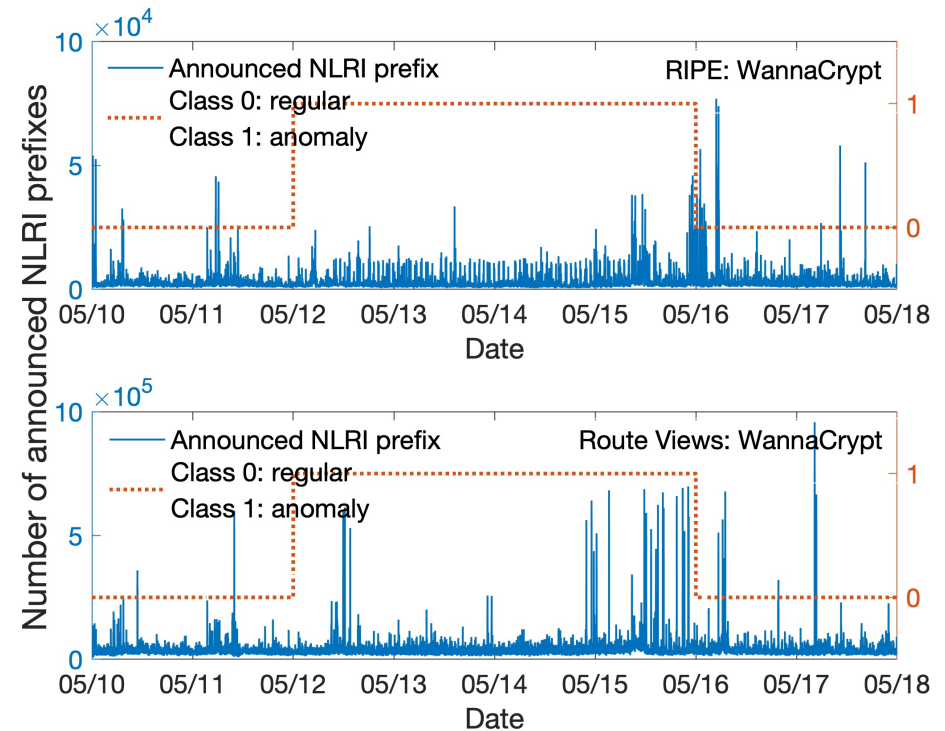
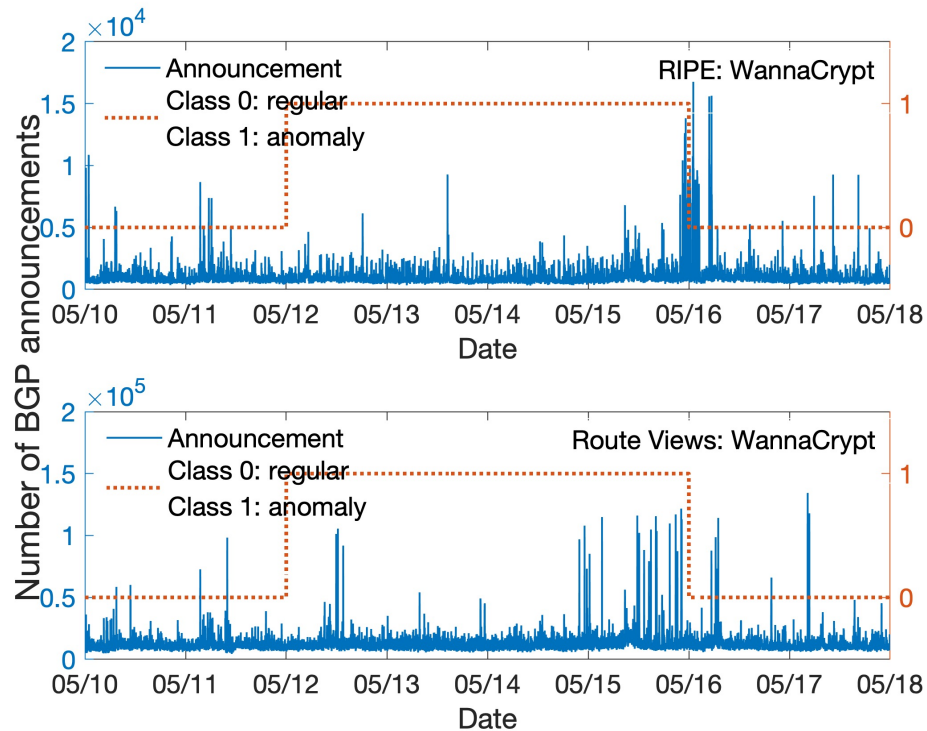
- Number of BGP announcements and announced NLRI prefixes vs. date:



BGP: Border Gateway Protocol
NLRI: Network Layer Reachability Information

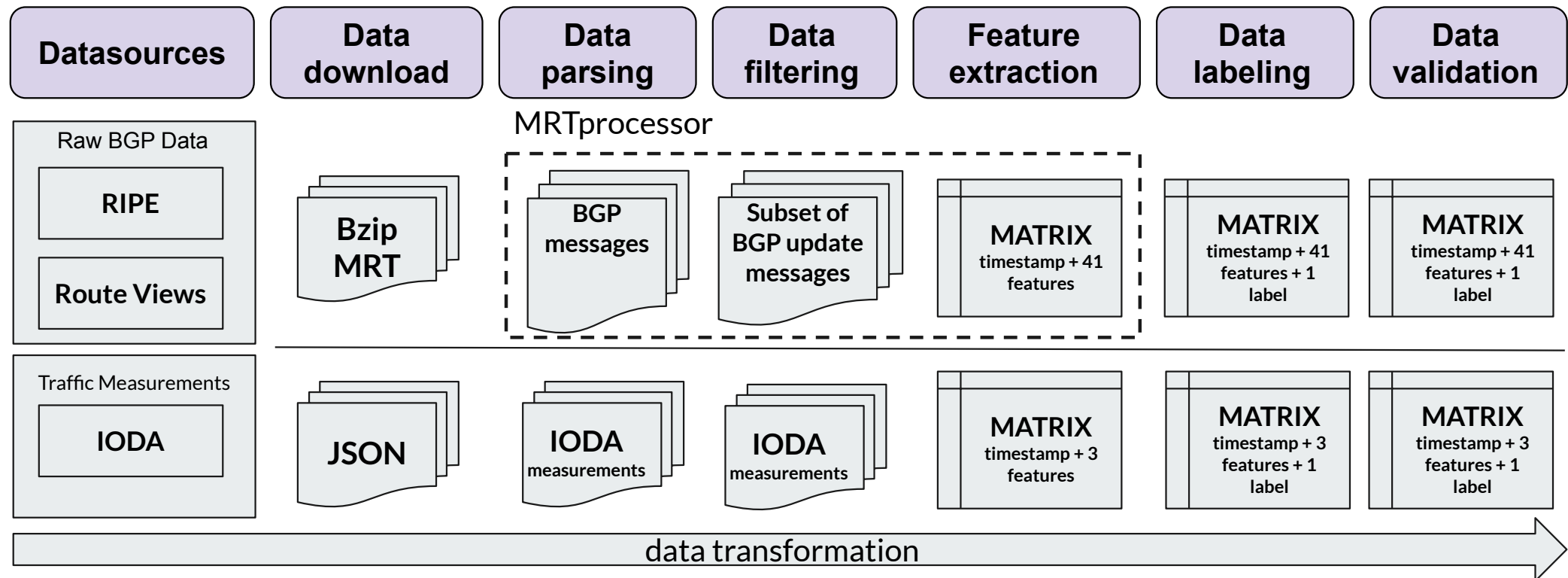
BGP Dataset: WannaCrypt (2017)

- Number of BGP announcements and announced NLRI prefixes vs. date:



BGP: Border Gateway Protocol
NLRI: Network Layer Reachability Information

Datasets

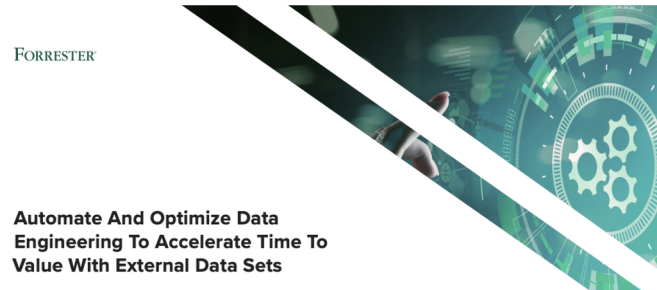


Data format from raw data to the final matrix with features and labels after the data labeling step. The IODA dataset collection site performs the data filtering step in advance, with filters already applied to the downloaded data.

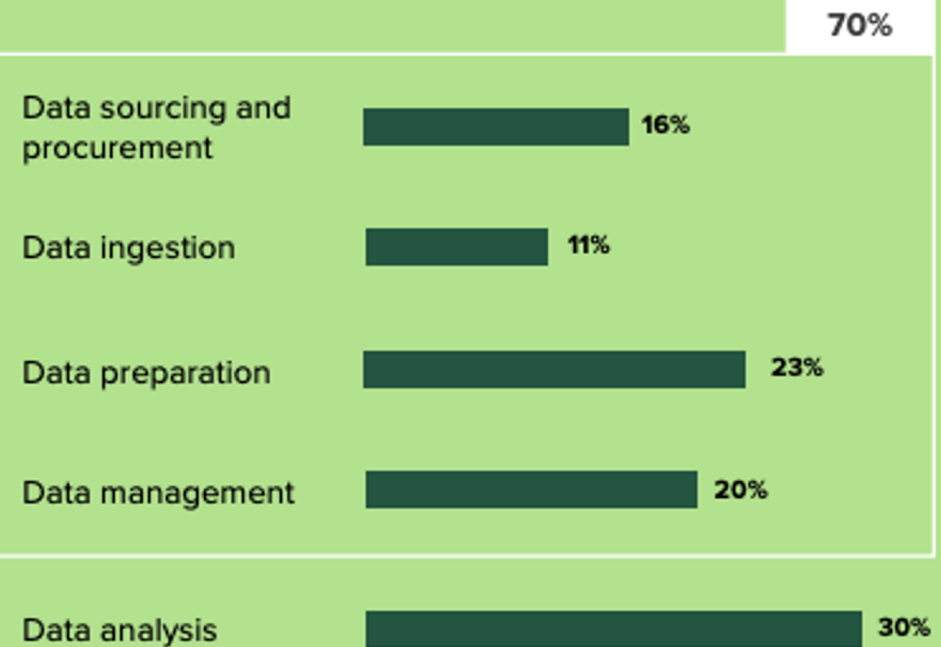
Generated Dataset: Sample

```
POSIXTIME, DATETIME, HOUR, MINUTE, SECOND, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21,
1692493200, 2023-08-20 01:00:00, 1, 0, 0, 127, 2, 133, 12, 9, 12, 9, 66, 5397, 0, 12, 2.1, 5, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
Col 2: POSIXTIME -08-20 01:01:00, 1, 1, 0, 204, 1, 204, 4, 9, 12, 9, 95, 18488, 0, 7, 3.4, 4, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1692493320, 2023-08-20 01:02:00, 1, 2, 0, 18, 0, 18, 0, 9, 10, 9, 2, 134, 0, 4, 0.3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 18,
1692493380, 2023-08-20 01:03:00, 1, 3, 0, 138, 0, 138, 0, 9, 11, 9, 134, 8402, 0, 5, 2.3, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1692493440, 2023-08-20 01:04:00, 1, 4, 0, 153, 2, 153, 2, 9, 11, 9, 76, 7156, 0, 8, 2.5, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
1692493500, 2023-08-20 01:05:00, 1, 5, 0, 158, 2, 187, 2, 8, 10, 8, 35, 4407, 0, 10, 2.6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
1692493560, 2023-08-20 01:06:00, 1, 6, 0, 135, 2, 137, 5, 9, 12, 9, 81, 5617, 0, 12, 2.2, 5, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
1692493620, 2023-08-20 01:07:00, 1, 7, 0, 126, 1, 126, 9, 9, 11, 9, 67, 6851, 0, 11, 2.1, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
1692493680, 2023-08-20 01:08:00, 1, 8, 0, 2, 0, 2, 0, 9, 9, 9, 0, 1, 0, 3, 0.0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1,
1692493740, 2023-08-20 01:09:00, 1, 9, 0, 95, 1, 95, 1, 9, 11, 9, 63, 3952, 0, 5, 1.6, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1692493800, 2023-08-20 01:10:00, 1, 10, 0, 125, 1, 125, 2, 9, 11, 9, 42, 6994, 0, 6, 2.1, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1692493860, 2023-08-20 01:11:00, 1, 11, 0, 218, 3, 220, 1293, 8, 12, 8, 48, 8296, 0, 12, 3.7, 6, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
2, 1692493920, 2023-08-20 01:12:00, 1, 12, 0, 292, 12, 292, 6883, 8, 11, 8, 92, 14761, 0, 11, 5.0, 6, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
3, 1692493980, 2023-08-20 01:13:00, 1, 13, 0, 465, 1, 520, 2, 8, 15, 8, 157, 21262, 0, 15, 7.8, 7, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
4, 1692494040, 2023-08-20 01:14:00, 1, 14, 0, 82, 0, 82, 0, 8, 10, 8, 5, 756, 0, 9, 1.4, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
5, 1692494100, 2023-08-20 01:15:00, 1, 15, 0, 111, 1, 111, 1, 9, 11, 9, 90, 4693, 0, 10, 1.9, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
6, 1692494160, 2023-08-20 01:16:00, 1, 16, 0, 114, 0, 122, 0, 9, 12, 9, 33, 4348, 0, 12, 1.9, 5, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
7, 1692494220, 2023-08-20 01:17:00, 1, 17, 0, 36, 1, 36, 37, 9, 10, 9, 4, 493, 0, 10, 0.6, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
```


How long does it take to create a dataset?



Where Are Organizations Spending Their Time With External Data Sets?



Source: <https://info.cruxinformatix.com/hubfs/Forrester-External-Data-Research-2023.pdf>

Roadmap

- Introduction
- Data collections
- **Feature selection**
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

Feature Selection

- Using feature selection algorithms to select the most relevant features in the original dataset often improves classification performance
- Various feature selection algorithms are used to reduce redundancies by ranking and identifying the most relevant features:
 - Fisher
 - minimum redundancy maximum relevance (mRMR)
 - mutual information base (MIBASE)
 - odds ratio (OR)
 - decision trees
 - extremely randomized trees (extra-trees)

Feature Selections: Extra-Trees

- The Gini importance is used to compute feature scores in a given dataset:

$$Importance(\mathbf{X}_c) = \frac{1}{N_T} \sum_T \sum_{t \in T: v(s_t) = \mathbf{X}_c} p(t) \Delta i(s_t, t),$$

where:

- $\mathbf{X}_c$ is the subset of $\mathbf{X}$ corresponding to one feature
- N_T is the number of trees
- t is the index of a node in a tree
- s_t is the direction of the split
- $v(s_t)$ is a randomly generated threshold
- $p(t)$ is the weight
- $\Delta i(st, t)$ is the decrease of the node impurity equivalent to its importance

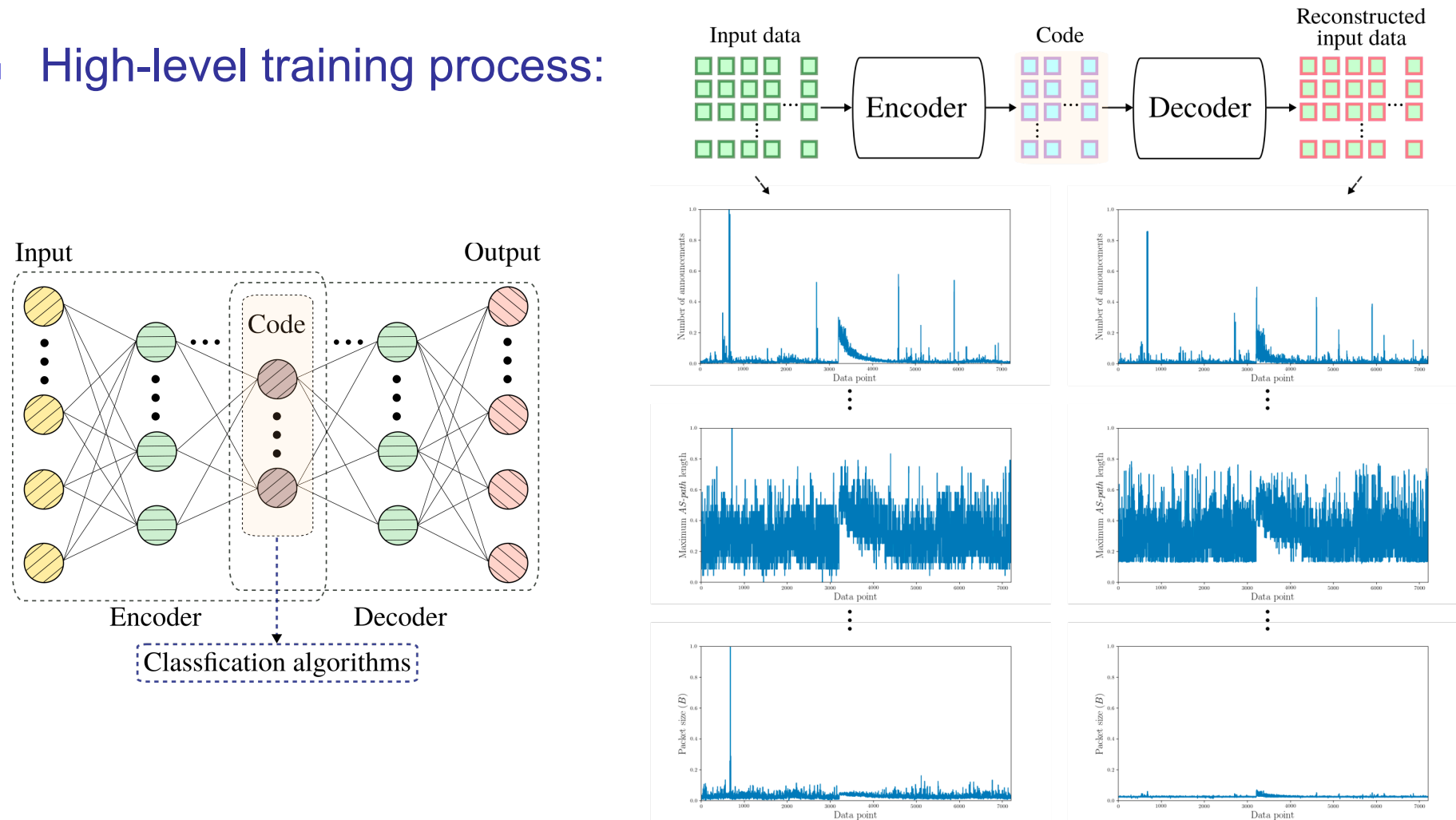
Dimension Reduction

- Dimension reduction (unsupervised learning):
 - using unlabeled input data to train a model
- Transform original data into the lower dimensional data that preserve characteristics of the original data:
 - autoencoders: unsupervised neural networks used to learn a representation from a given dataset
- Deep autoencoder with various LSTM/GRU hidden layers used for dimension reduction

LSTM: Long short-term memory
GRU: Gated recurrent unit

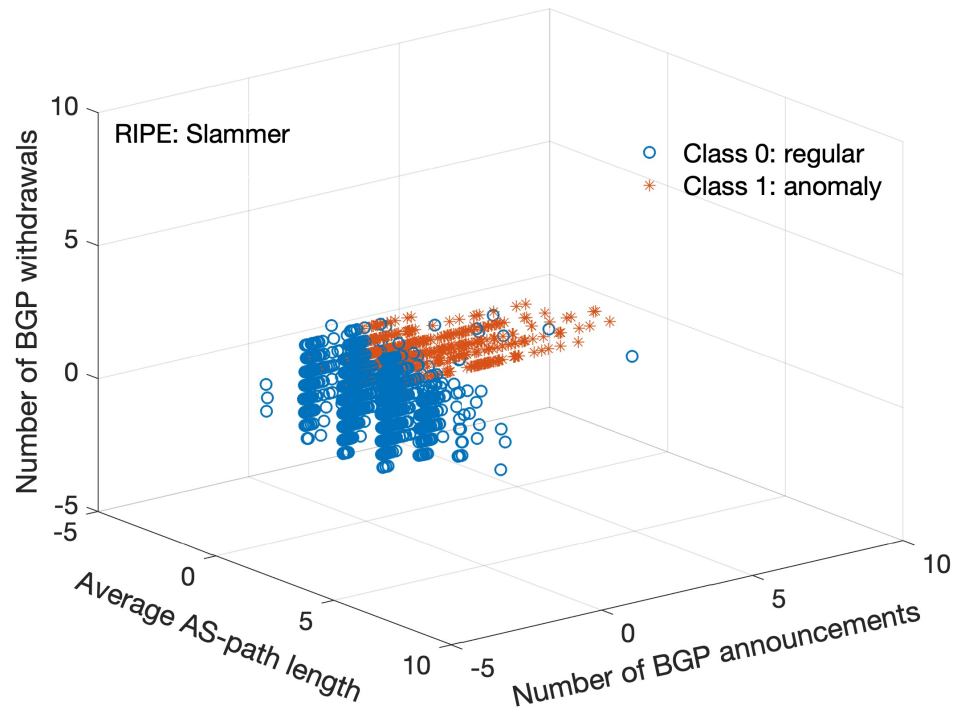
Autoencoders

- High-level training process:



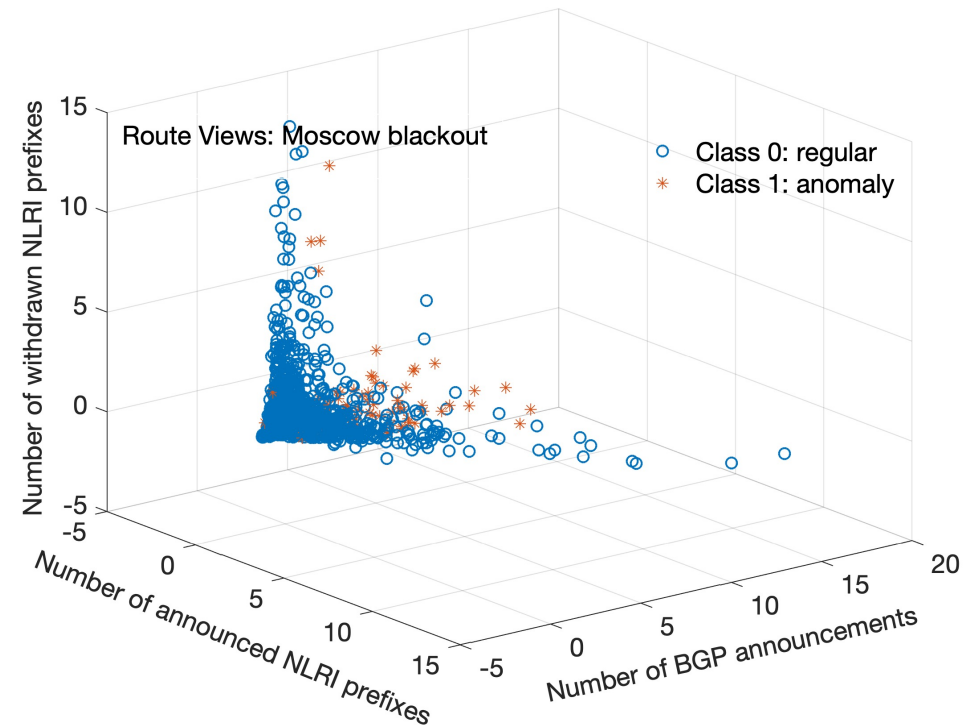
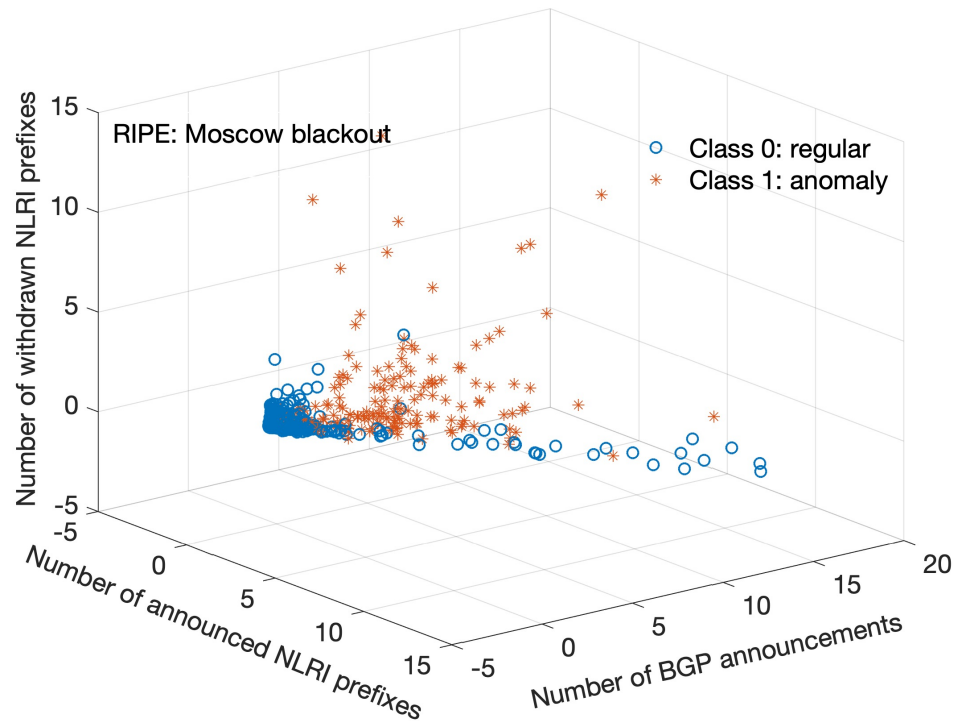
BGP Dataset: Slammer (2003)

- Average AS-path length vs. number of BGP announcements vs. number of BGP withdrawals:



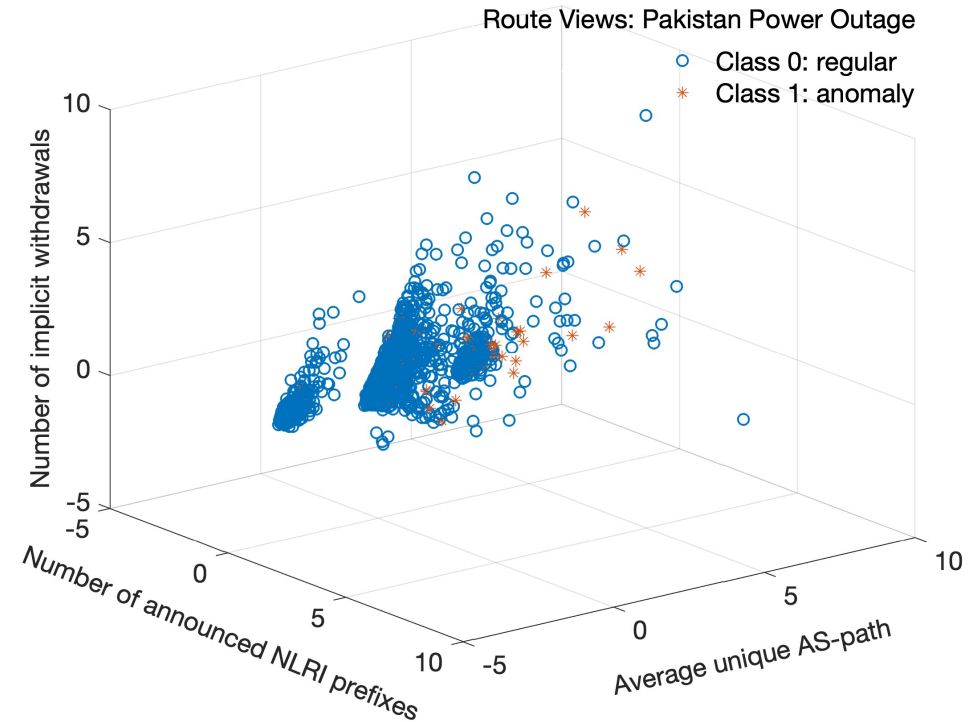
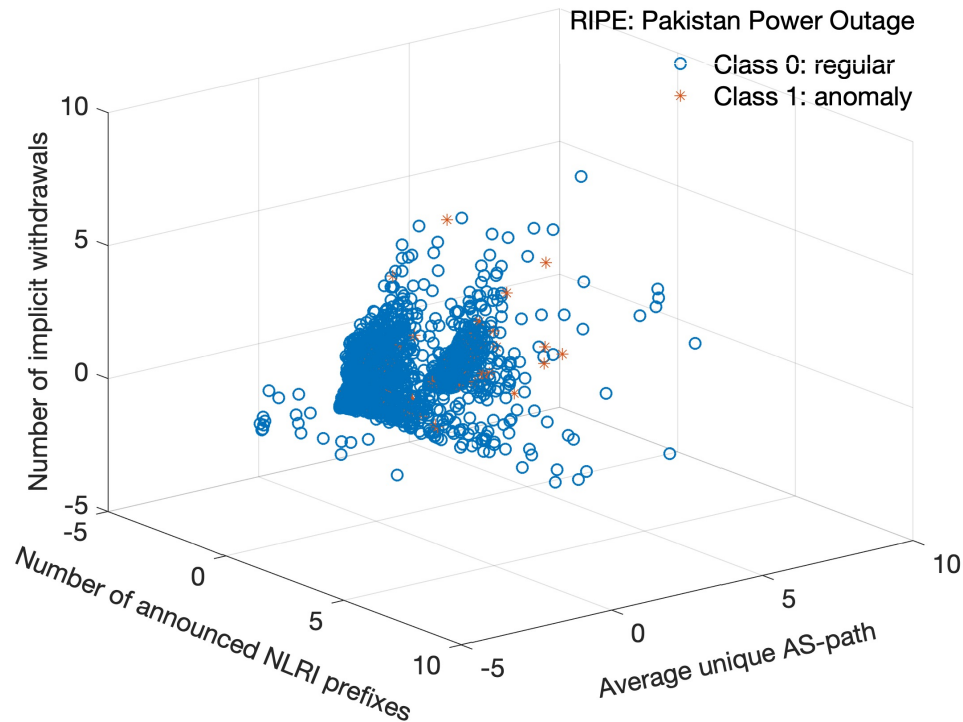
BGP Dataset: Moscow Blackout (2005)

- Number of announced NLRI prefixes vs. number of BGP announcements vs. number of withdrawn NLRI prefixes:



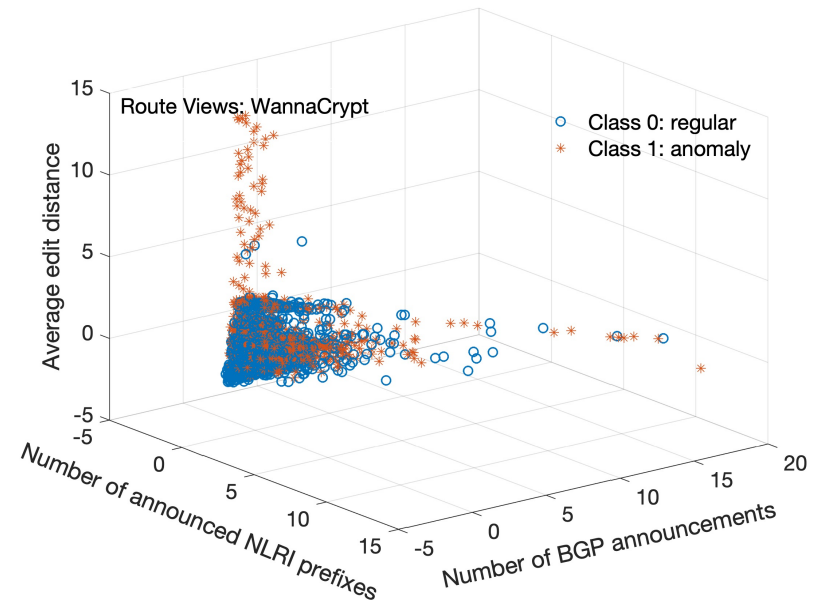
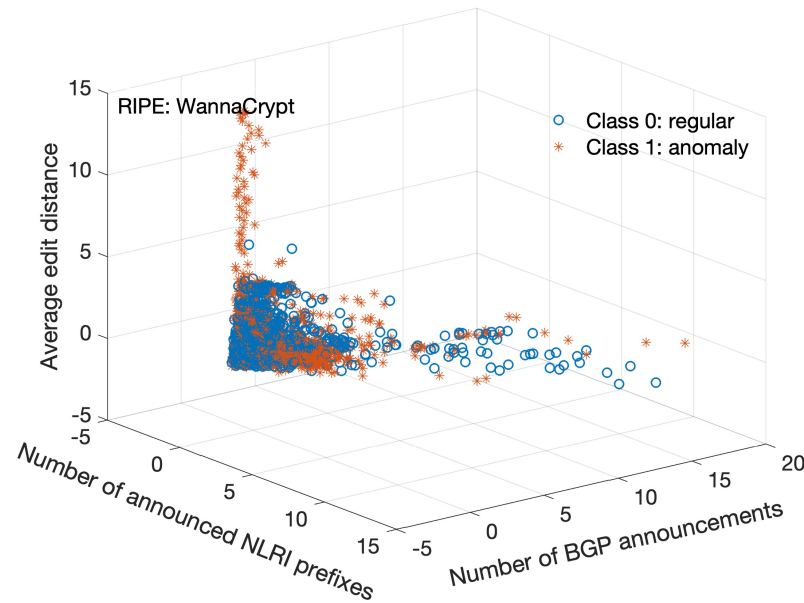
BGP Dataset: Pakistan Power Outage (2021)

- Number of announced NLRI prefixes vs. average unique AS-path vs. number of implicit withdrawals:



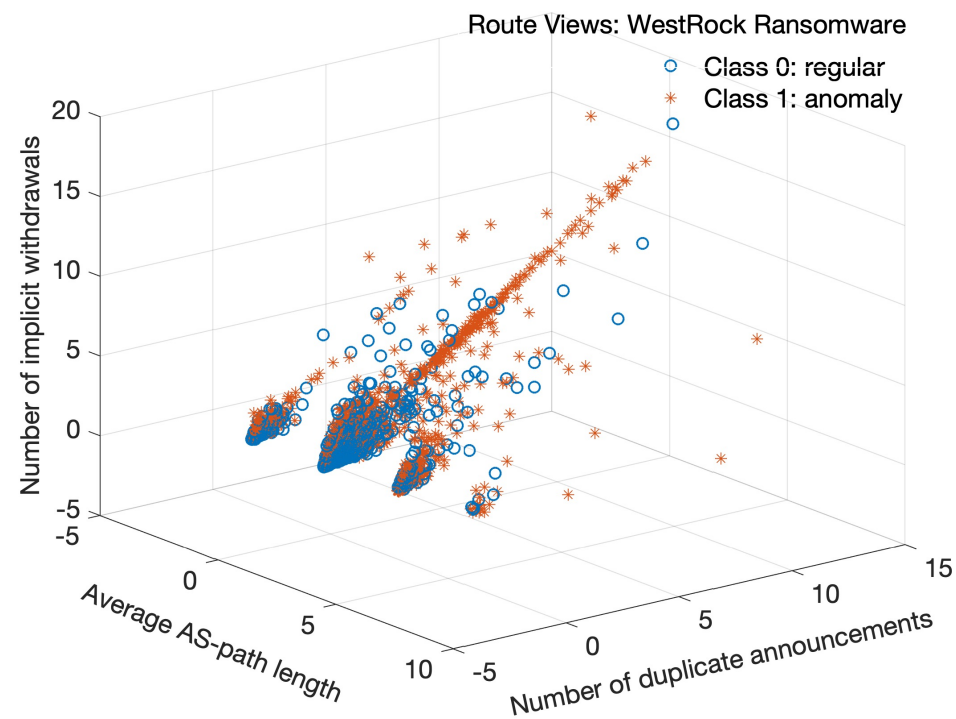
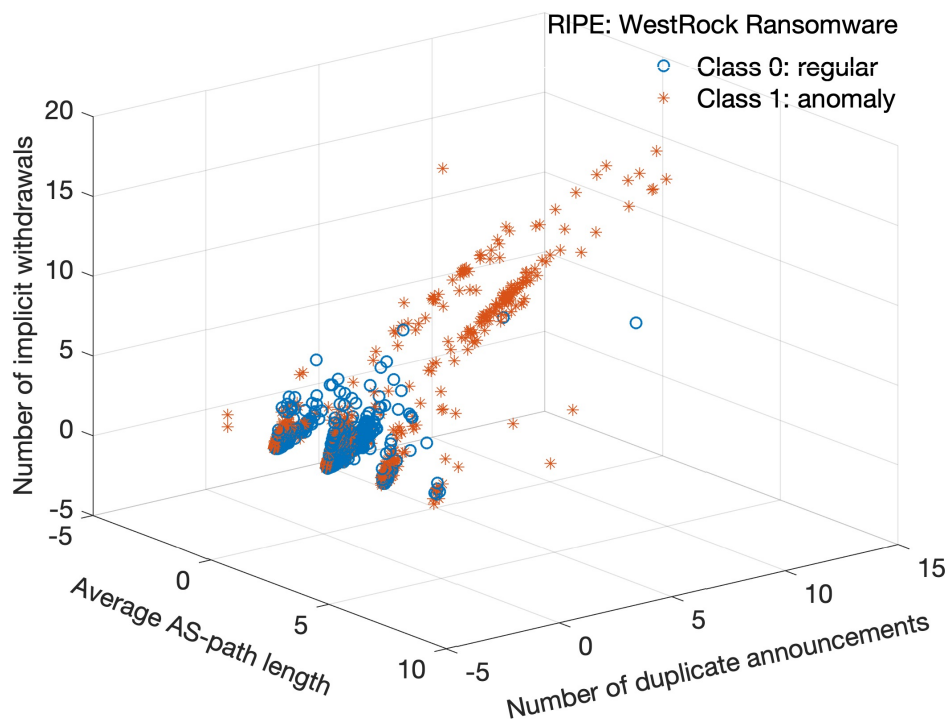
BGP Dataset: WannaCrypt (2017)

- Number of announced NLRI prefixes vs. number of BGP announcements vs. average edit distance:



BGP dataset: WestRock (2021)

- Average unique AS-path vs. number of duplicate announcements vs. number of implicit withdrawals:



Roadmap

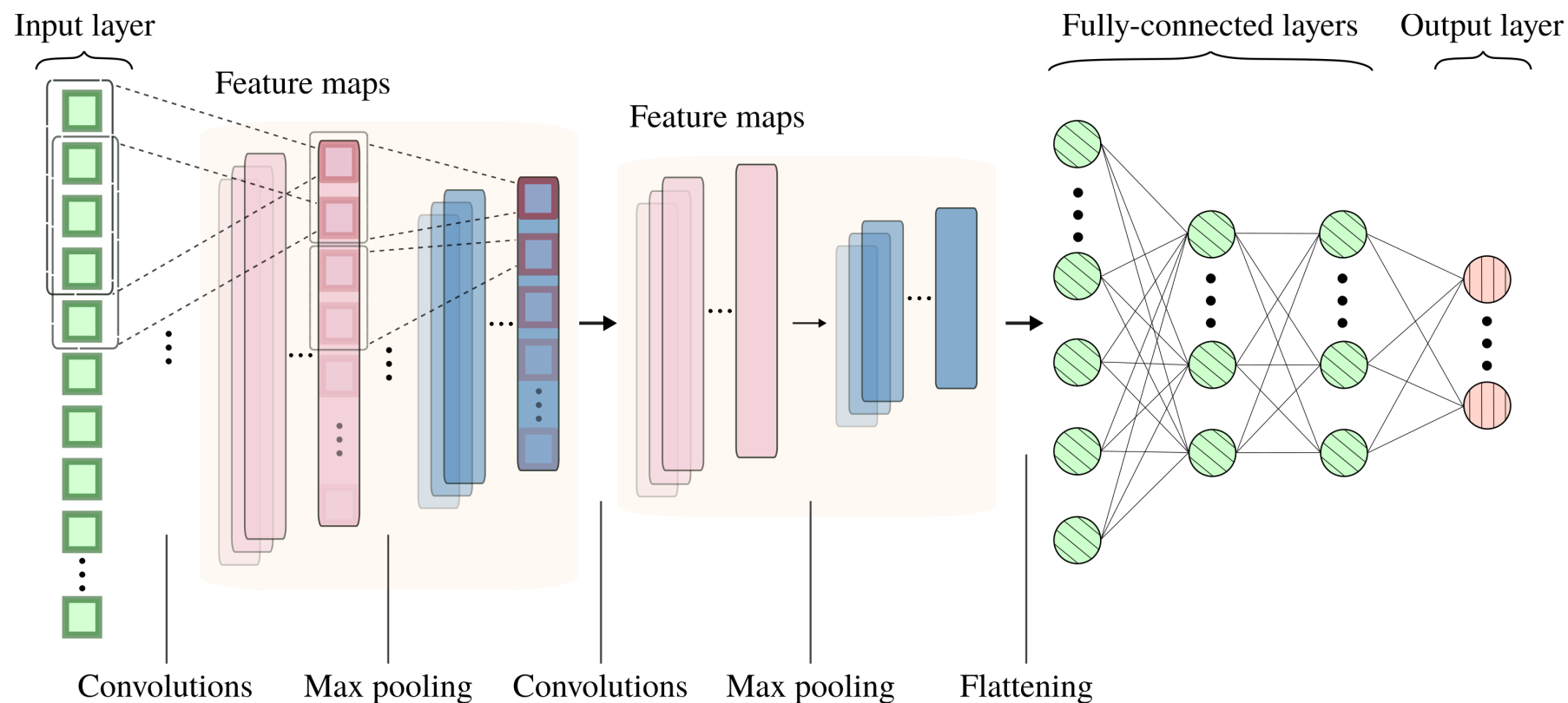
- Introduction
- Data collections
- Feature selection
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosted decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

Machine Learning Algorithms

- Supervised machine learning algorithms:
 - Support vector machine (**SVM**)
 - Convolutional neural networks (**CNN**)
 - Recurrent neural networks (**RNN**):
 - Long short-term memory: **LSTM** and **Bi-LSTM**
 - Gated recurrent unit: **GRU** and **Bi-GRU**
 - Broad learning system: **BLS** and its extensions
 - Gradient Boosted Decision Trees (**GBDT**):
 - **XGBoost**
 - **LightGBM**
 - **CatBoost**

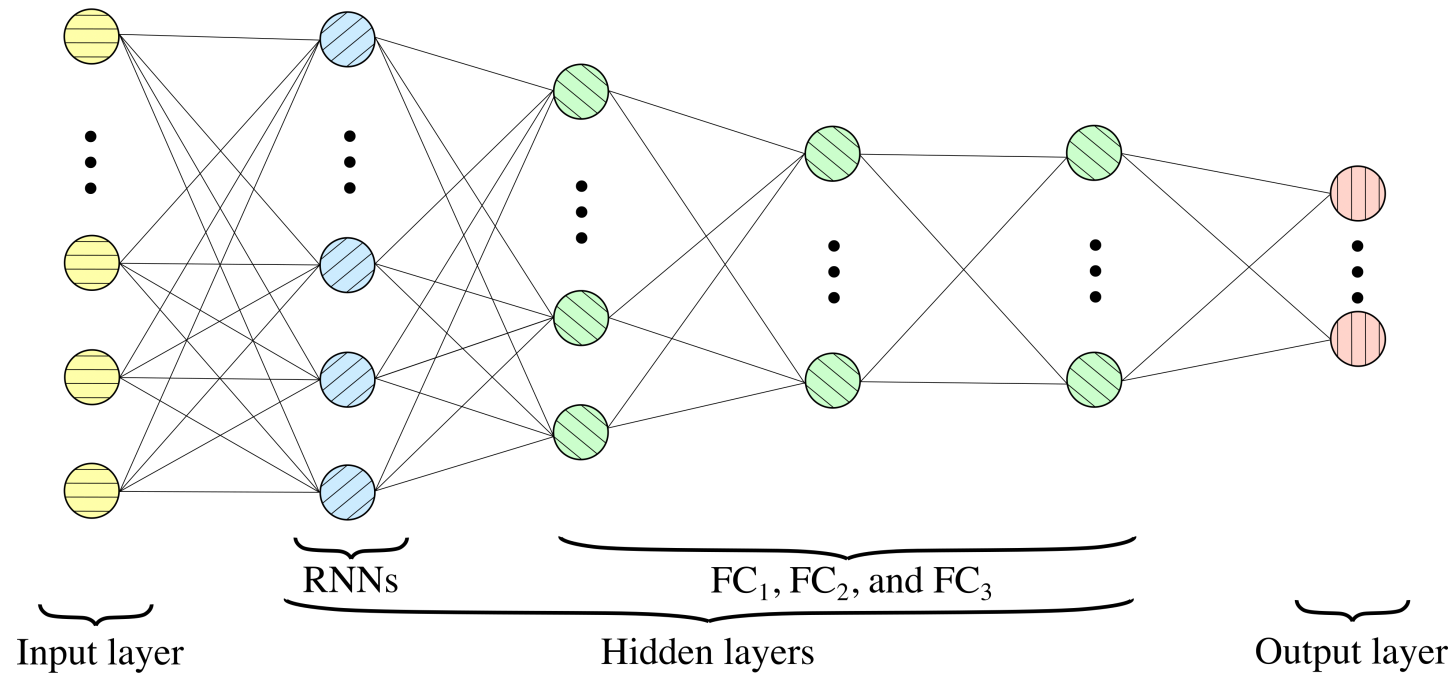
Convolutional Neural Network

- The high-level structure of a CNN using 1-dimensional input data:



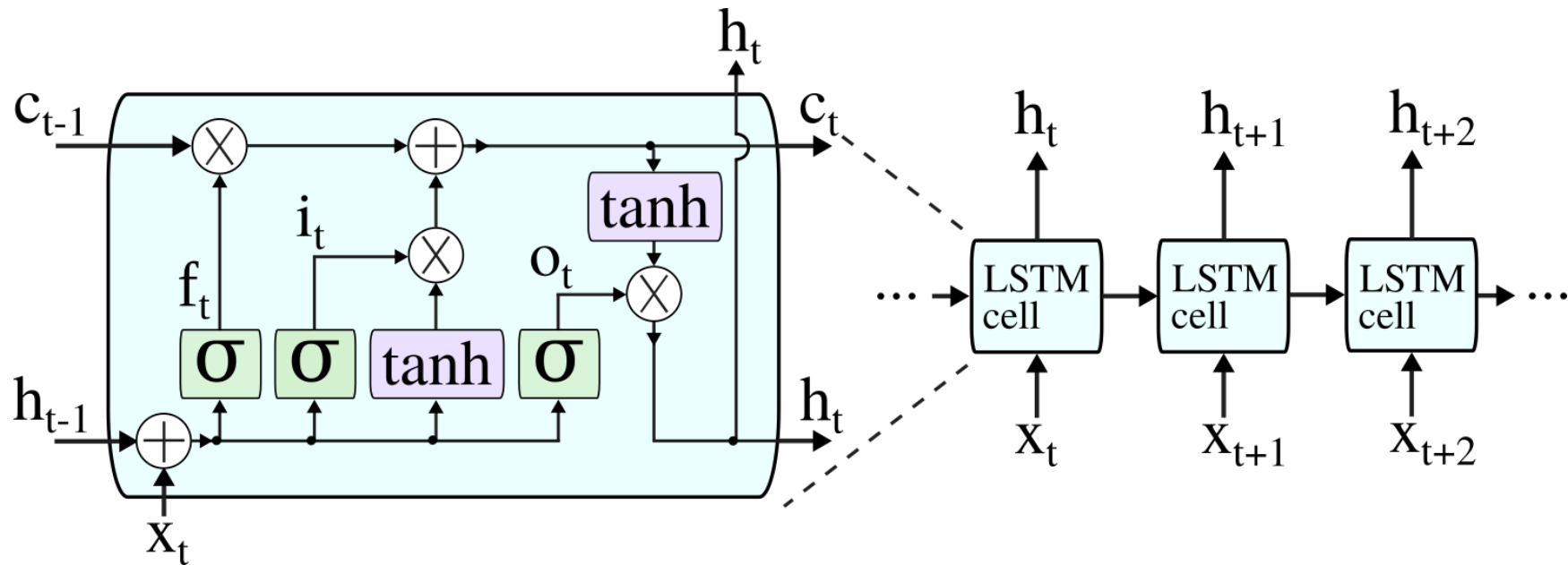
Deep Learning Neural Network

- 37 RNNs, 80 FC_1 , 32 FC_2 , and 16 FC_3 fully connected (FC) hidden nodes:



Long Short-Term Memory: LSTM

- Repeating module for the LSTM neural network:



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.

Long Short-Term Memory: LSTM

- The outputs of the forget gate f_t , the input gate i_t , and the output gate o_t at time t are:

$$f_t = \sigma(W_{if}x_t + b_{if} + U_{hf}h_{t-1} + b_{hf})$$

$$i_t = \sigma(W_{ii}x_t + b_{ii} + U_{hi}h_{t-1} + b_{hi})$$

$$o_t = \sigma(W_{io}x_t + b_{io} + U_{ho}h_{t-1} + b_{ho}),$$

where:

$\sigma(\cdot)$: logistic sigmoid function

x_t : current input vector

h_{t-1} : previous output vector

W_{if} , U_{hf} , W_{ii} , U_{hi} , W_{io} , and U_{ho} : weight matrices

b_{if} , b_{hf} , b_{ii} , b_{hi} , b_{io} , and b_{ho} : bias vectors

Long Short-Term Memory: LSTM

- Output i_t of the input gate decides if the information will be stored in the cell state. The sigmoid function is used to update the information.

- Cell state c_t :

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_{ic}x_t + b_{ic} + U_{hc}h_{t-1} + b_{hc}),$$

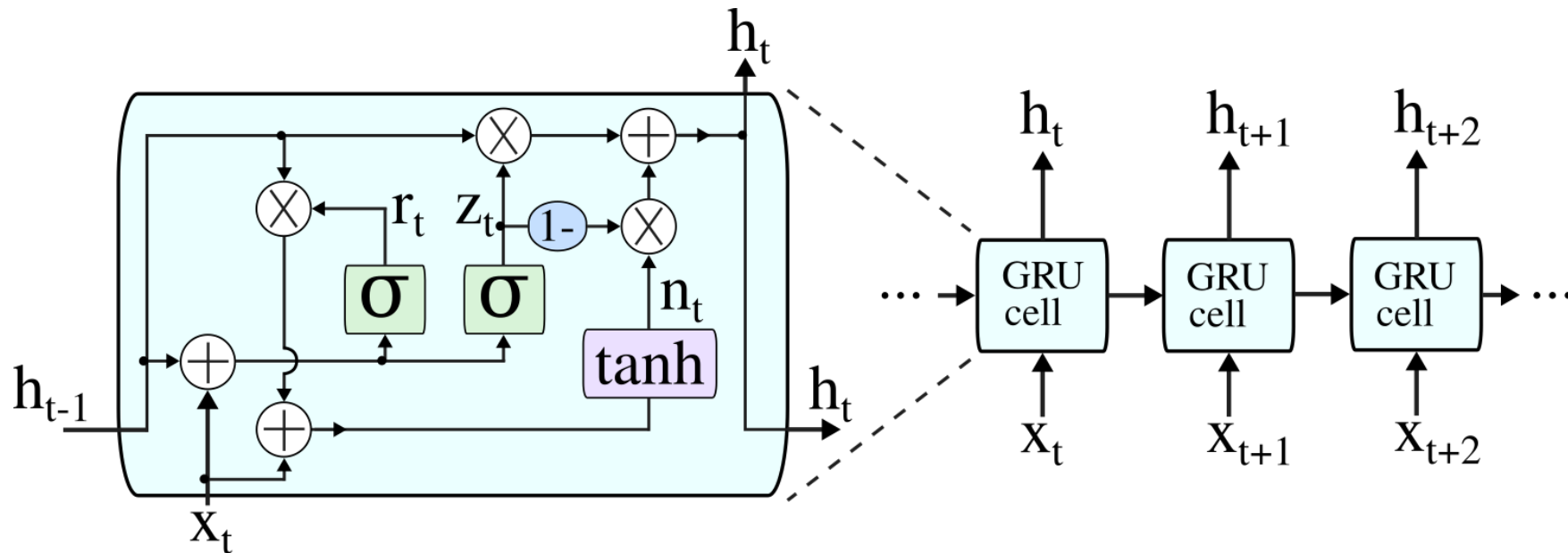
where:

- $*$ denotes element-wise multiplications
- $\tanh$ function: used to create a vector for the next cell state
- Output of the LSTM cell:

$$h_t = o_t * \tanh(c_t)$$

Gated Recurrent Unit: GRU

- Repeating module for the GRU neural network:



K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translations," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

Gated Recurrent Unit: GRU

- The outputs of the reset gate r_t and the update gate z_t at time t :

$$r_t = \sigma(W_{ir}x_t + b_{ir} + U_{hr}h_{t-1} + b_{hr})$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + U_{hz}h_{t-1} + b_{hz}),$$

where:

- σ : sigmoid function
- x_t : input
- h_{t-1} is the previous output of the GRU cell
- W_{ir} , U_{hr} , W_{iz} , and U_{hz} : weight matrices
- b_{ir} , b_{hr} , b_{iz} , and b_{hz} : bias vectors

Gated Recurrent Unit: GRU

- Output of the GRU cell:

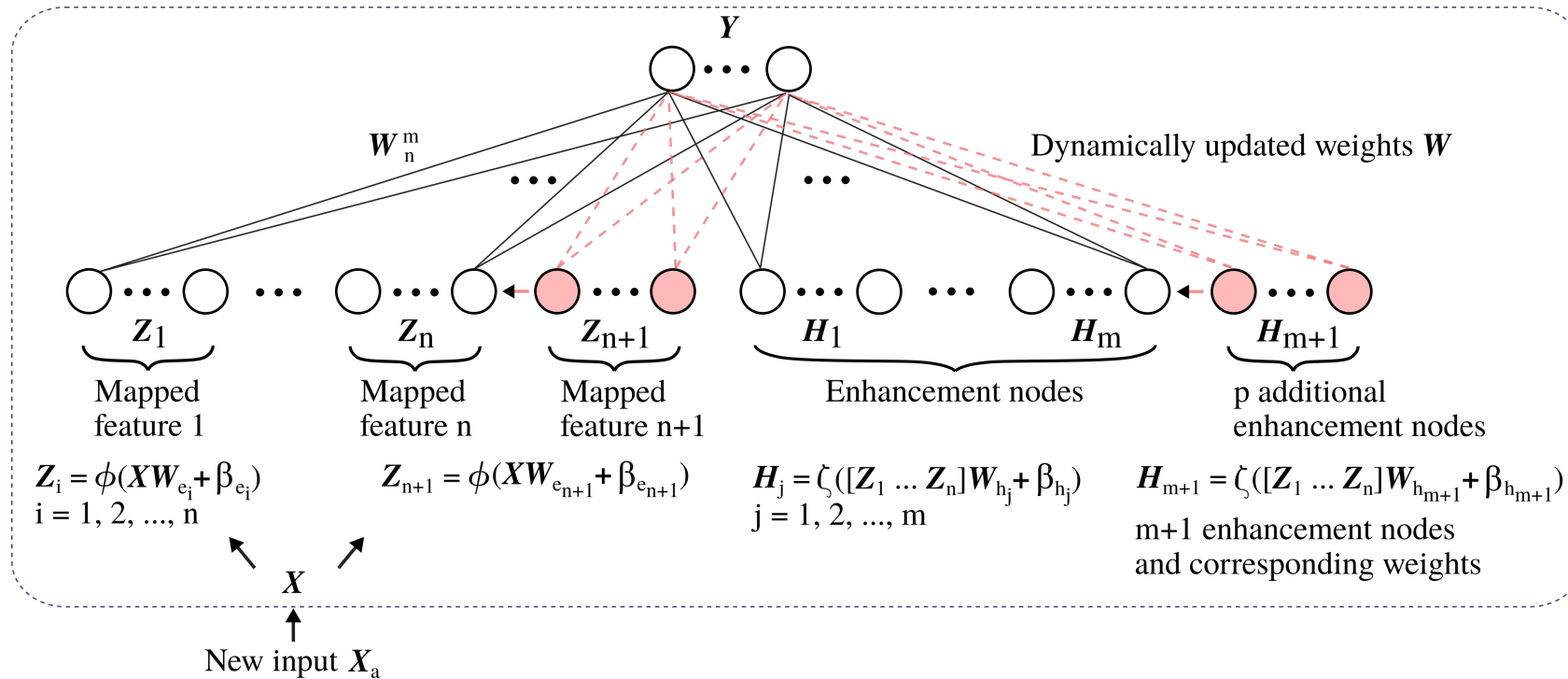
$$h_t = (1 - z_t) * n_t + z_t * h_{t-1},$$

where n_t :

- $n_t = \tanh(W_{in}x_t + b_{in} + r_t * (U_{hn}h_{t-1} + b_{hn}))$
- W_{in} and U_{hn} : weight matrices
- b_{in} and b_{hn} : bias vectors

Broad Learning System

- Broad Learning System (BLS) algorithm with increments of mapped features, enhancement nodes, and new input data:



C. L. P. Chen and Z. Liu, "Broad learning system: an effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

Original BLS

- State matrix $\mathbf{A}_x$ is constructed from groups of mapped features $\mathbf{Z}^n$ and groups of enhancement nodes $\mathbf{H}^m$ as:

$$\begin{aligned}\mathbf{A}_x &= [\mathbf{Z}^n \mid \mathbf{H}^m] \\ &= \left[\phi(\mathbf{X}\mathbf{W}_{e_i} + \boldsymbol{\beta}_{e_i}) \mid \xi(\mathbf{Z}_x^n \mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j}) \right], \\ &\quad i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, m,\end{aligned}$$

where:

- ϕ and ξ : projection mappings
- $\mathbf{W}_{e_i}, \mathbf{W}_{h_j}$: weights
- $\boldsymbol{\beta}_{e_i}, \boldsymbol{\beta}_{h_j}$: bias parameters

Original BLS

- Moore-Penrose **pseudo inverse** of matrix A_x is computed to calculate the weights of the output:

$$W_n^m = [A_n^m]^+ Y$$

- Calculated using **ridge regression**:

$$W_n^m = [(A_n^m)^T A_n^m + \lambda I]^{-1} (A_n^m)^T Y$$

- During the testing process, data labels are deduced using the calculated weights W_n^m , mapped features Z_n , and enhancement nodes H_m :

$$\begin{aligned} Y &= A_n^m W_n^m \\ &= [Z_1, \dots, Z_n | H_1, \dots, H_m] W_n^m \end{aligned}$$

- Modified to include additional **mapped features** Z_{n+1} , **enhancement nodes** H_{m+1} , and/or **input nodes** X_a

RBF-BLS Extension

- The **RBF function** is implemented using Gaussian kernel:

$$\xi(x) = \exp\left(-\frac{\|x - c\|^2}{\gamma^2}\right)$$

- Weight vectors of the output $\mathbf{HW}$ are deduced from:

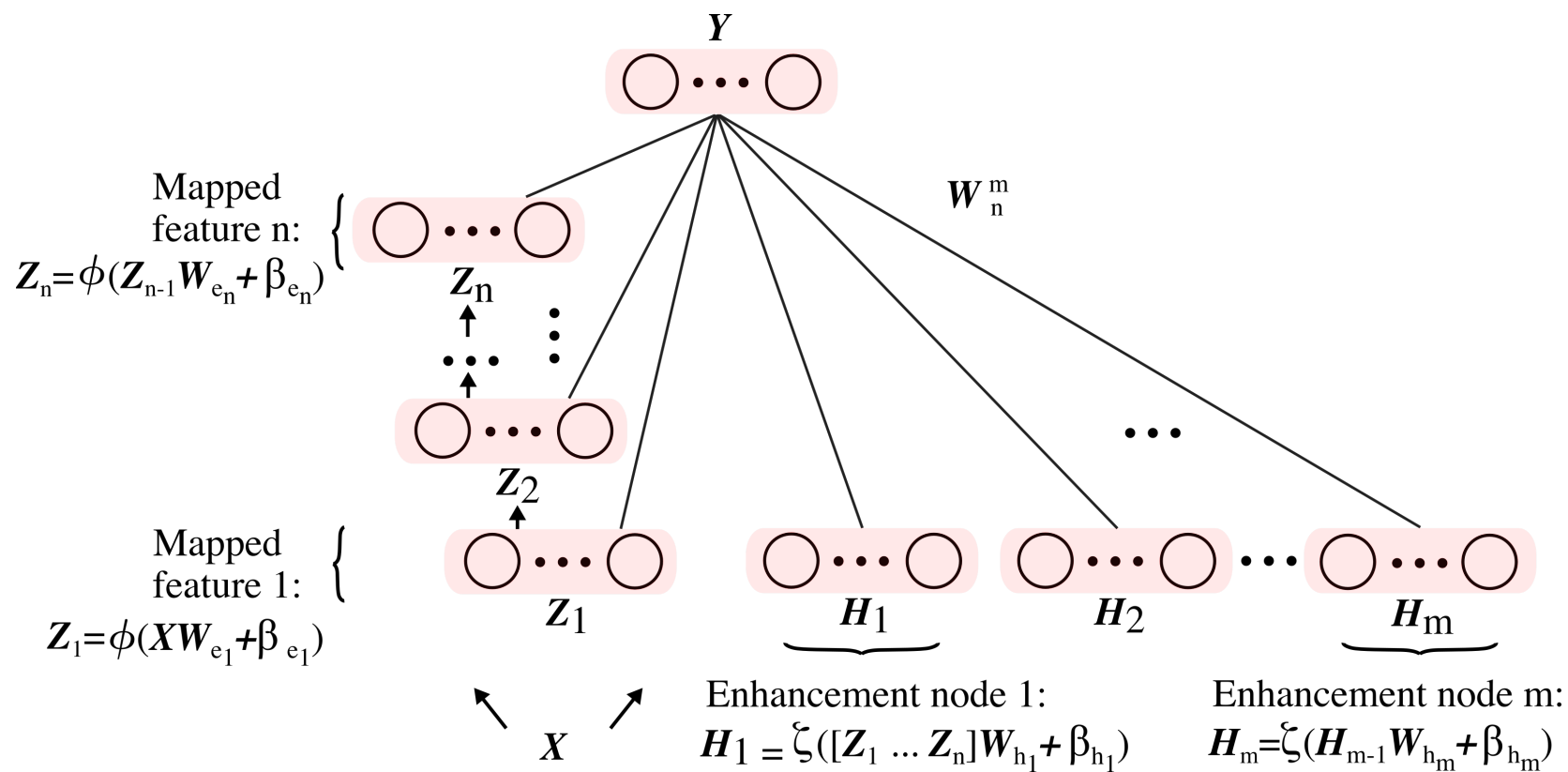
$$\begin{aligned}\mathbf{W} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y} \\ &= \mathbf{H}^+ \mathbf{Y},\end{aligned}$$

where:

- $\mathbf{W} = [\omega_1, \omega_2, \dots, \omega_k]$: output weights
- $\mathbf{H} = [\xi_1, \xi_2, \dots, \xi_k]$: hidden nodes
- $\mathbf{H}^+$: pseudoinverse of $\mathbf{H}$

RBF: Radial basis function
BLS: Broad learning system

Cascades of Mapped Features

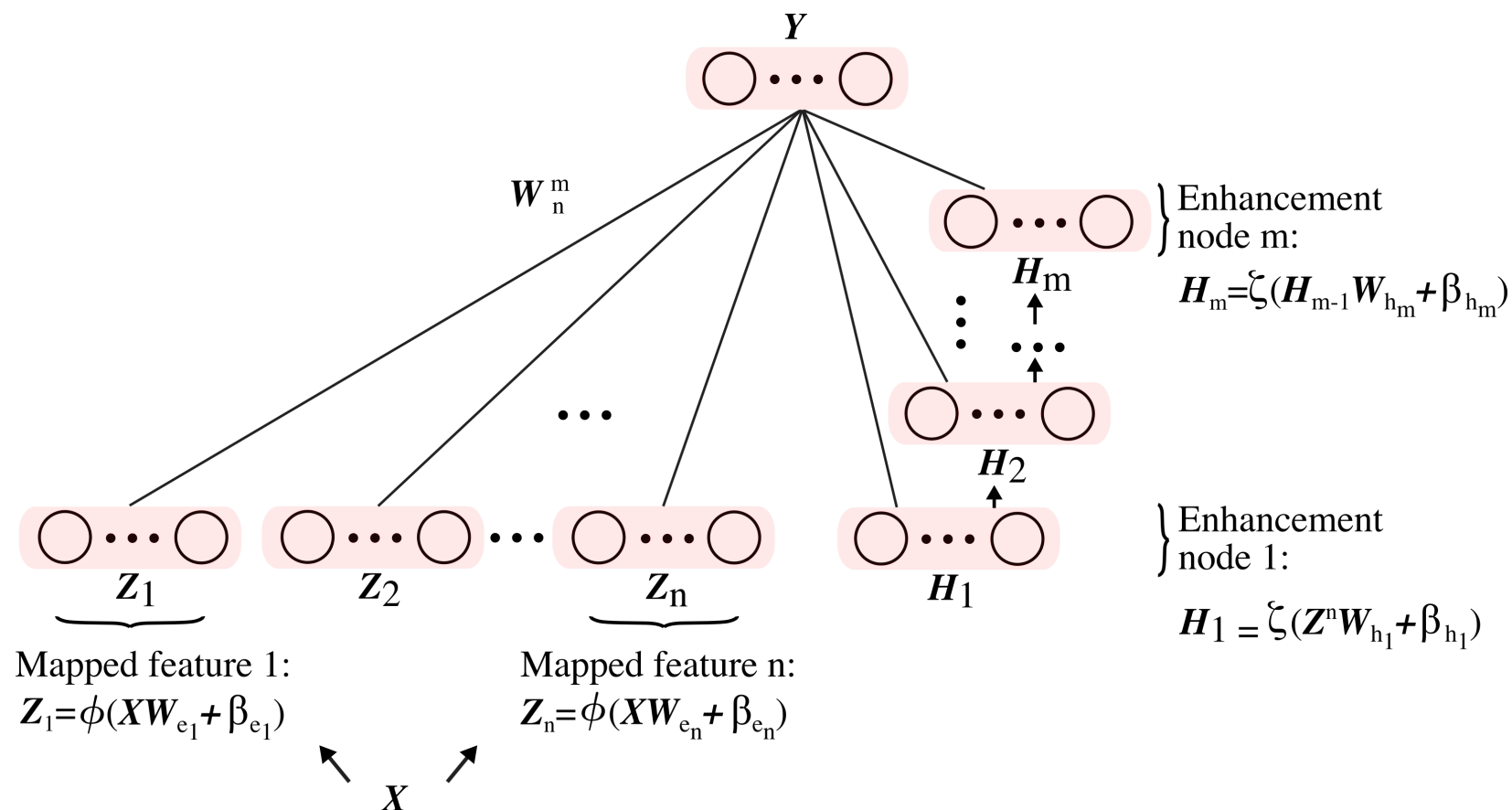


Cascades of Mapped Features

- **Cascade of mapped features (CFBLS):**
the new group of mapped features is created by using the previous group ($k - 1$).
- Groups of mapped features are formulated as:

$$\begin{aligned} \mathbf{Z}_k &= \phi(\mathbf{Z}_{k-1} \mathbf{W}_{e_k} + \boldsymbol{\beta}_{e_k}) \\ &\triangleq \phi^k(\mathbf{X}; \{\mathbf{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^k), \text{ for } k = 1, \dots, n \end{aligned}$$

Cascades of Enhancement Nodes



Cascades of Enhancement Nodes

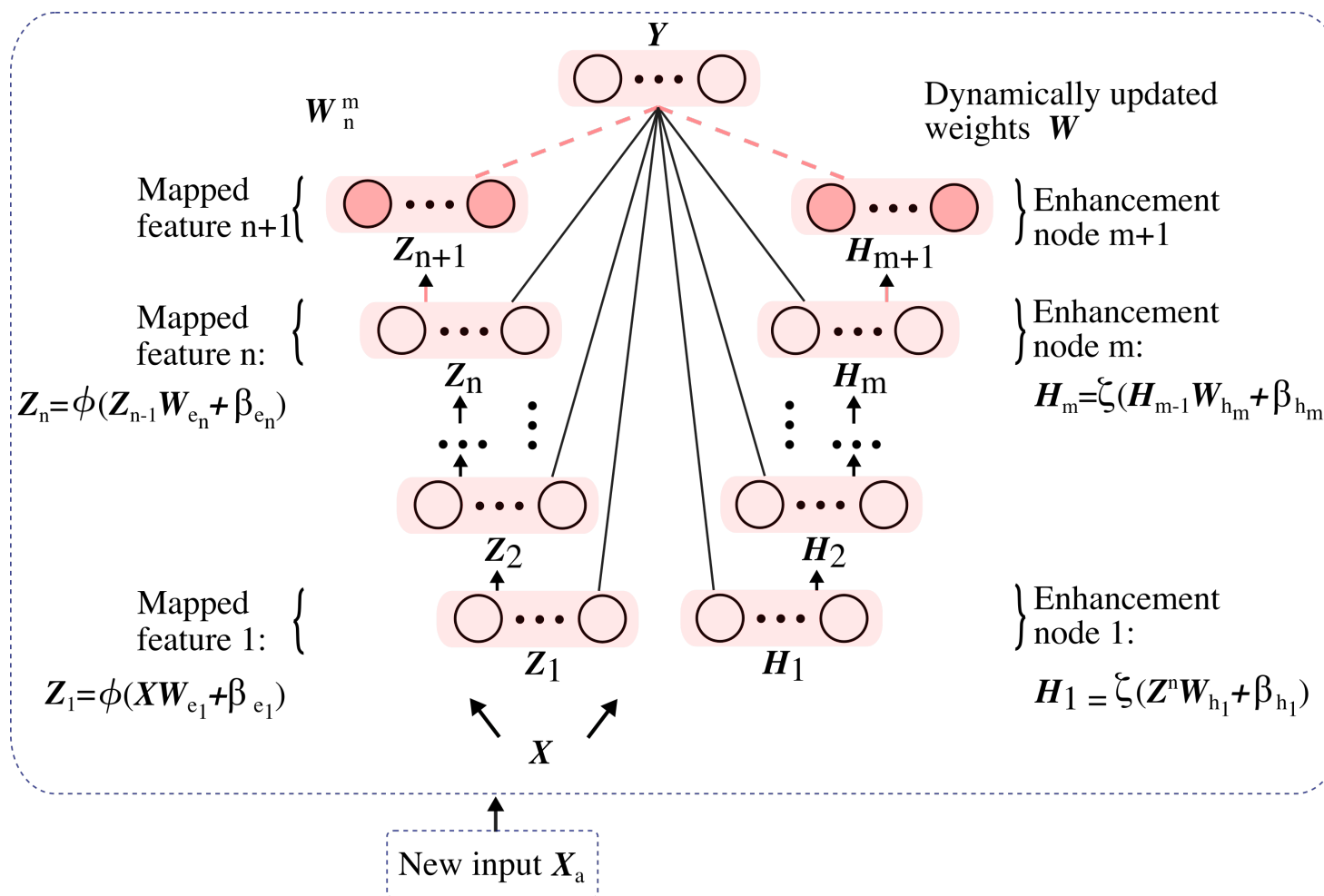
- The first enhancement node in **cascade of enhancement nodes (CEBLS)** is generated from mapped features.
- The subsequent enhancement nodes are generated from previous enhancement nodes creating a cascade:

$$\mathbf{H}_u \triangleq \xi^u \left(\mathbf{Z}^n ; \{ \mathbf{W}_{h_i}, \boldsymbol{\beta}_{h_i} \}_{i=1}^u \right), \text{ for } u = 1, \dots, m,$$

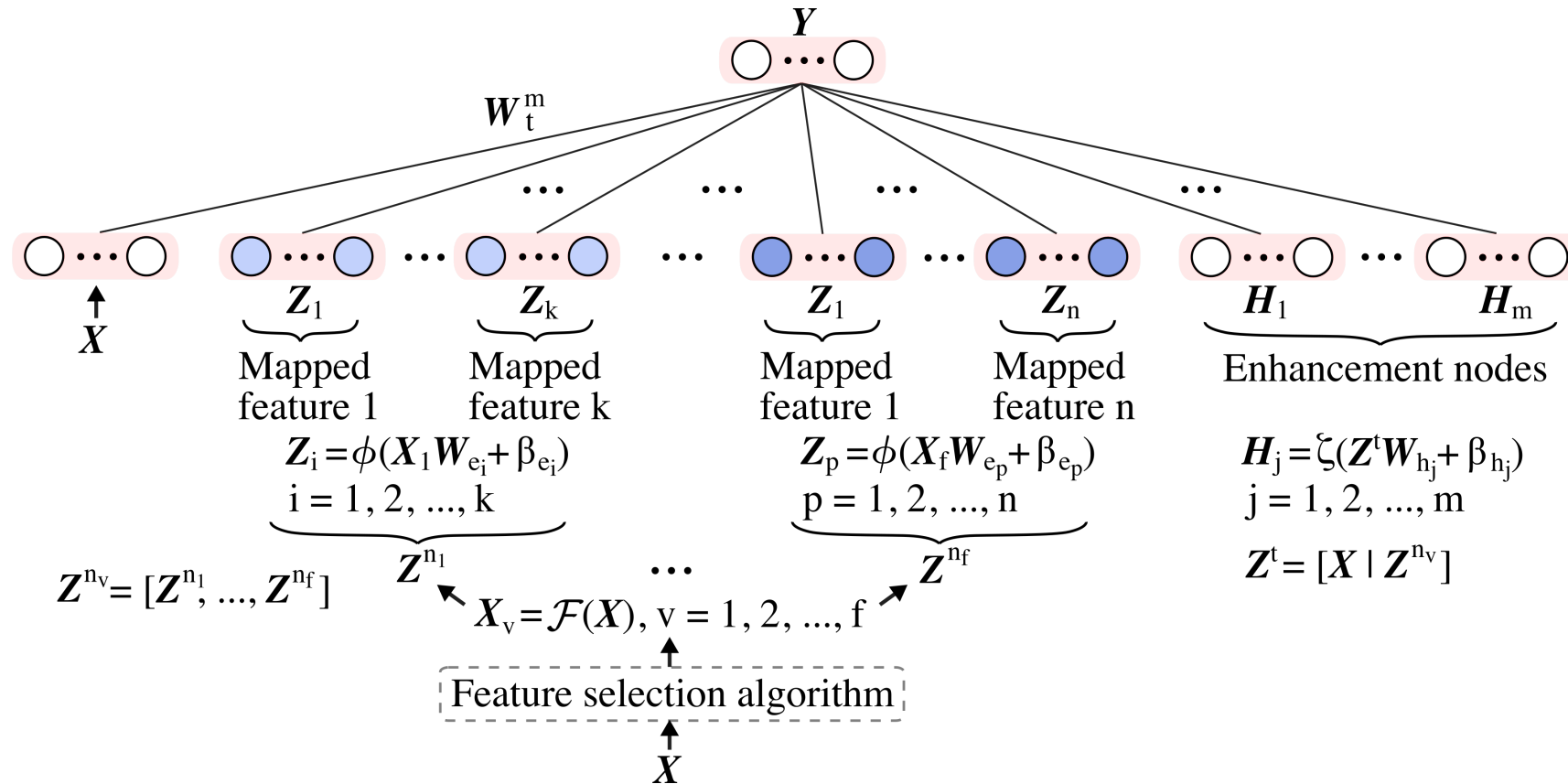
where:

- $\mathbf{W}_{h_i}$ and $\boldsymbol{\beta}_{h_i}$: randomly generated

Cascades with Incremental Learning



Variable Features Broad Learning System: VFBL



Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, "Machine learning for detecting anomalies and intrusions in communication networks," *IEEE JSAC*, vol. 39, no. 7, pp. 2254-2264, July 2021.

Variable Features Broad Learning System: VFBLS

- Subsets of input data $\mathbf{X}$ using a feature selection algorithm $\mathcal{F}$:

$$\mathbf{X}_v = \mathcal{F}(\mathbf{X}), \quad v = 1, 2, \dots, f$$

- Sets of groups of mapped features: $\mathbf{Z}^{n_v} = [\mathbf{Z}^{n_1}, \dots, \mathbf{Z}^{n_f}]$
- Concatenation of $\mathbf{X}$ and $\mathbf{Z}^{n_v}$: $\mathbf{Z}^t = [\mathbf{X} \mid \mathbf{Z}^{n_v}]$
- Enhancement nodes:

$$\mathbf{H}_j = \xi \left(\mathbf{Z}^t \mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j} \right), \quad j = 1, 2, \dots, m$$

where:

- f : number of subsets
- n_v : number of sets of mapped features
- ξ : projection mapping

Variable Features Broad Learning System: VFBL

- State matrix A_t^m : concatenation of Z^t and H^m
- Ridge regression algorithm is employed to compute the weights W_t^m based on A_t^m and given labels Y
- Error function, minimized during the training process:

$$E(W_t^m) = (\|W_t^m - Y\|_2)^2 + (\lambda \|W_t^m\|_2)^2$$

- Output weights:

$$W_t^m = (\lambda I + (A_t^m)^T A_t^m)^{-1} (A_t^m)^T Y$$

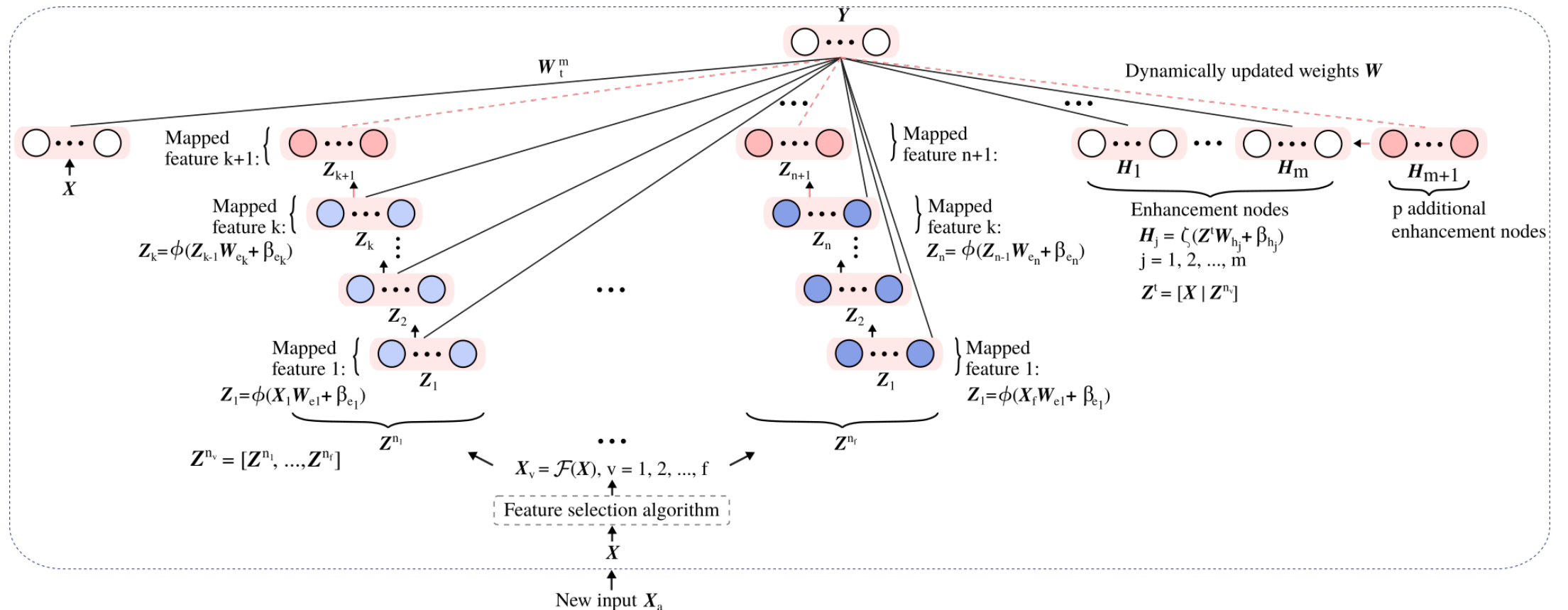
where:

- λ is the sparse regularization coefficient

Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, "Machine learning for detecting anomalies and intrusions in communication networks," *IEEE JSAC*, vol. 39, no. 7, pp. 2254-2264, July 2021.

Variable Features Broad Learning System: VCFBLS

■ Incremental learning:



Gradient Boosting Decision Trees

- Gradient Boosting Machines (**GBM**):
 - Boosting algorithms that employ functional gradient descent to minimize the loss function
- **GBDT**: **GBM** variant that employs decision trees as estimators algorithms:
 - **XGBoost**: eXtreme gradient boosting
 - **LightGBM**: light gradient boosting machine
 - **CatBoost**: categorical boosting

Gradient Boosting Decision Trees: GBDT

- When training a GBDT model with K estimators using N data points, the predicted output for the i^{th} data point $\mathbf{x}_i$ is:

$$\hat{y}_i = \sum_{k=1}^K f_k(\mathbf{x}_i),$$

where:

- f_k : the k^{th} decision tree
- $\mathbf{x}_i$: a row vector of matrix $\mathbf{X}$ containing input data and represents one collection sample

Gradient Boosting Decision Trees: GBDT

- In the k^{th} iteration, predicted output is evaluated using the k^{th} estimator and k^{th} decision tree:

$$\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(\mathbf{x}_i),$$

where:

- $\hat{y}_i^{(k)}$: predicted output of the i^{th} data point
- $\hat{y}_i^{(k-1)}$: previously predicted output
- f_k : the k^{th} decision tree

Gradient boosting decision trees: GBDT

- Goal of the GBDT models is to minimize the objective function:

$$\mathcal{L}^{(k)} = \sum_{i=1}^N l(y_i - \hat{y}_i^{(k)}) + \Omega(f_k),$$

where:

- $l(\cdot)$: loss function
- y_i : true value of the i^{th} data point
- $\hat{y}_i^{(k)}$: predicted output of the i^{th} data point for the k^{th} iteration
- $\Omega(f_k)$: (optional) regularization term

GBDT: XGBoost, LightGBM, CatBoost

- **XGBoost:**
 - adds an L^2 norm regularization term to avoid over-fitting
 - employs the second-order Taylor series to approximate its objective function
- **LightGBM:**
 - accelerate the training speed by using gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB)
- **CatBoost:**
 - deals with categorical features
 - employs target statistic to convert categorical to numerical features
 - employs ordered boosting

GBDT: XGBoost, LightGBM, CatBoost

- GBDT training time is affected by:
 - number of the iterations
 - learning rate
 - the maximum depth
 - number of leaves in the decision trees
 - the shortest training time: **LightGBM** models

GBDT: XGBoost

- Optimal solution of the objective function:

$$\mathcal{L}^{*(k)} = -\frac{1}{2} \sum_{t=1}^T \frac{(\sum_{i \in I_t} g_i)^2}{\sum_{i \in I_t} h_i + \lambda} + \gamma T$$

- This optimal value is used to evaluate the quality of a tree structure $q(X)$
- Tree structure with the lowest optimal value is selected for each iteration

GBDT: LightGBM

- In a decision tree, nodes are split based on features with the largest information gain, which depends on the variance gain $\tilde{V}_j$ for feature j computed after splitting as:

$$\begin{aligned}\tilde{V}_j(d) = & \frac{1}{N \times N_l^j(d)} \left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i \right)^2 \\ & + \frac{1}{N \times N_r^j(d)} \left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i \right)^2\end{aligned}$$

where:

- d : splitting point
- N : number of data points
- N_l^j and N_r^j : numbers of data points related to left and right child nodes
- g_i : gradient for data point x_i

GBDT: LightGBM

- The sampling ratios a and b are used to calculate the normalization coefficient:

$$\frac{1-a}{b}$$

- Subsets of $A(B)$:
 - $A_l(B_l)$: left child nodes
 - $A_r(B_r)$: right child nodes

GBDT: CatBoost

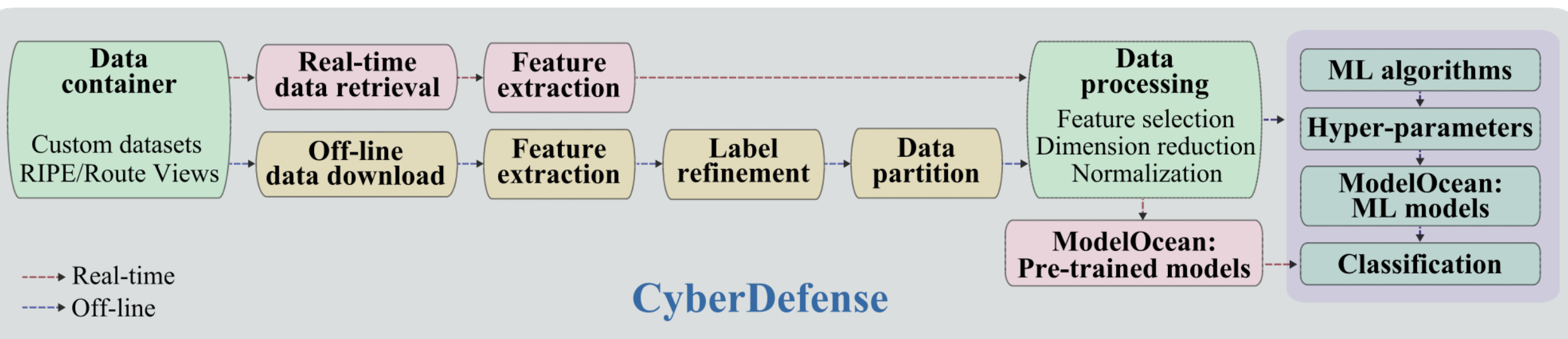
- **CatBoost**: introduced to deal with categorical features
- Employs the ordered boosting algorithm and offers an effective approach when compared to **XGBoost** and **LightGBM**
- Target statistic was used to convert categorical to numerical features while keeping the dimension of the dataset unchanged
- Ordered boosting addresses the prediction shift when building the decision trees during the training process
- Symmetric (oblivious) decision trees are used to avoid over-fitting and reduce the time required to grow the tree
- CatBoost offers plain and ordered boosting modes

Roadmap

- Introduction
- Data collections
- Feature selection
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

CyberDefense

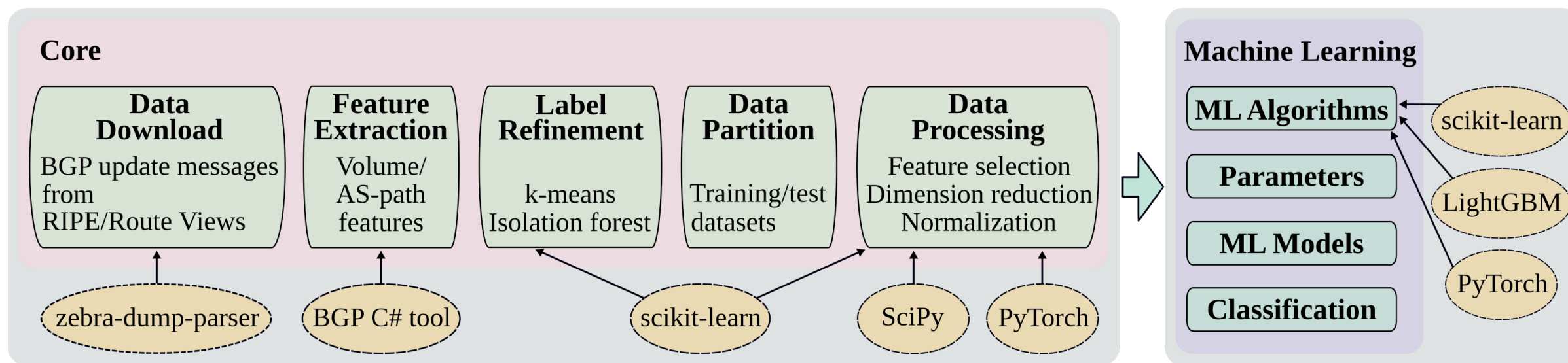
■ Architecture:



CyberDefense: Tool for Detecting Network Anomalies and Intrusions,
<https://github.com/zhida-li/cyberDefense>

BGPGuard

■ Architecture:

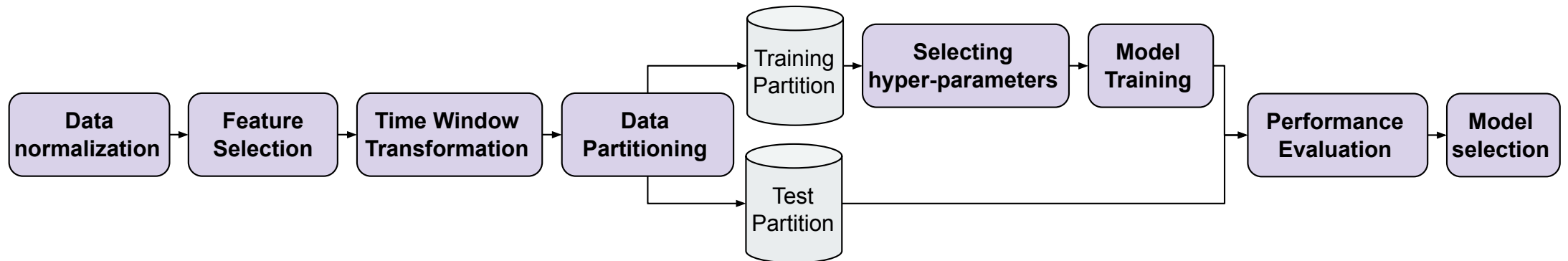


BGPGuard: a BGP Anomaly Detection Tool,
<https://github.com/zhida-li/BGPGuard>

Experimental Procedure

Machine learning pipeline:

- Pre-processing steps: data normalization, feature selection, time window transformation, and data partitioning.
- The best hyper-parameters: selected through an exhaustive grid search strategy.
- A model is trained and evaluated for each combination of hyper-parameters.
- The best model selected based on performance metrics.

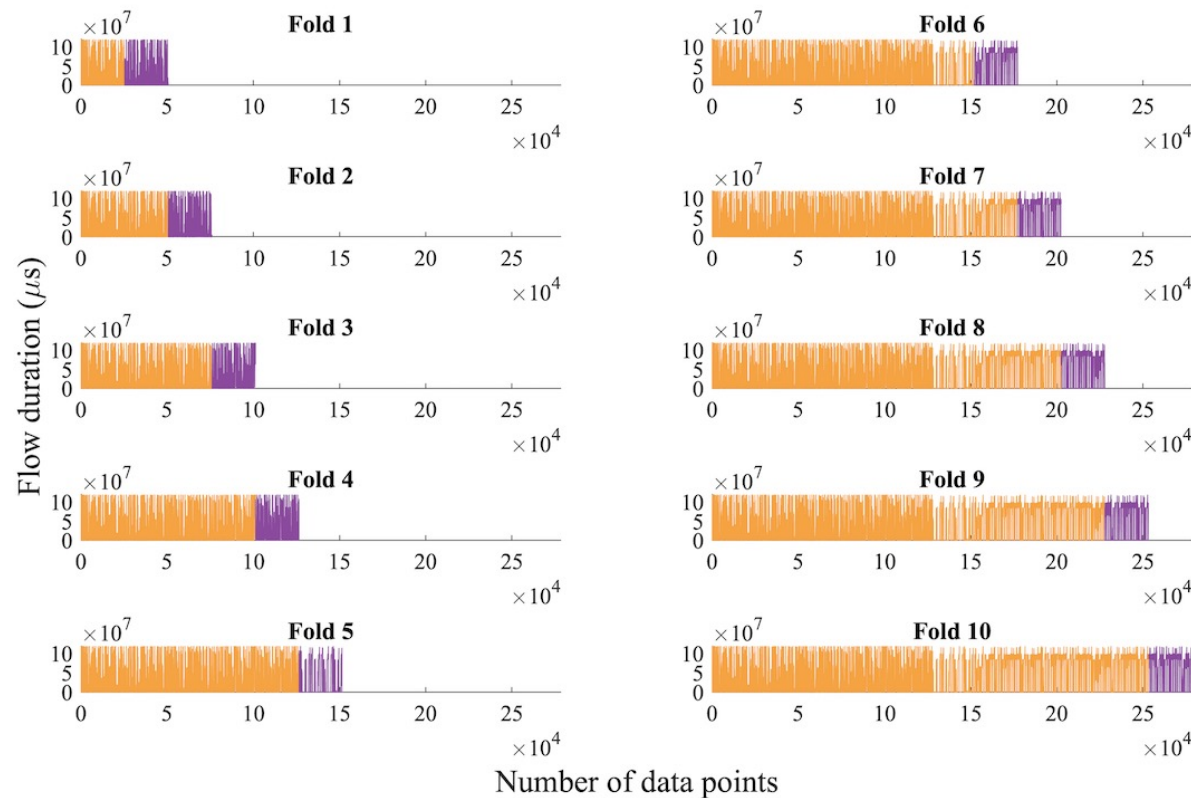


Experimental Procedure

- **Step 1:** Normalize training and test datasets.
- **Step 2:** Train the models using 10-fold validation and tune model parameters.
- **Step 3:** Test the **best** models.
- **Step 4:** Evaluate models based on:
 - accuracy
 - F-score
 - precision
 - sensitivity
 - confusion matrix
 - training time

Time Series Split for the 10-fold Cross-Validation

- Generations of training (orange) and validation (purple) datasets:



Data Normalization and Feature Selection

- **Data normalization:**
 - Uniform feature scales and distributions achieved using data normalization to ensure equal feature contributions.
 - Z-Score used to normalize datasets under consideration.
- **Feature selection:**
 - Features may have different relevance and some may be redundant irrelevant.
 - Selecting the most relevant features and their combinations may improve classification performance.
 - Scikit-learn implementation of extremely randomized trees (extra-trees) used for feature selection.

BGP Features

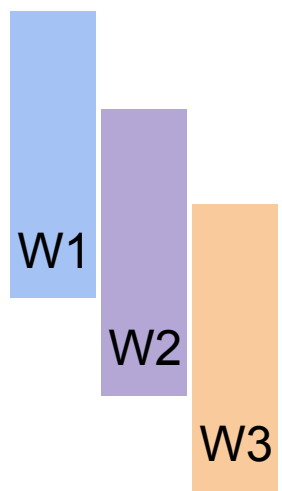
Feature ID	Feature name	Computational complexity
F1	Number of announcements	$O(n)$
F2	Number of withdrawals	$O(n)$
F3	Number of announced NLRI prefixes	$O(n)$
F4	Number of withdrawn NLRI prefixes	$O(n)$
F5	Average AS-path length	$O(n)$
F6	Maximum AS-path length	$O(n)$
F7	Average unique AS-path length	$O(n)$
F8	Number of duplicate announcements	$O(n^2)$
F9	Number of implicit withdrawals	$O(n^2)$
F10	Number of duplicate withdrawals	$O(n^2)$
F11	Maximum edit distance	$O(n^2)$
F12	Arrival rate	$O(n)$
F13	Average edit distance	$O(n^2)$
F14-F23	Maximum AS-path = [11-20]	$O(n)$
F24-F33	Maximum edit distance = [7-16]	$O(n)$

Feature ID	Feature name	Computational complexity
F34	Number of Interior Gateway Protocol (IGP) packets	$O(n)$
F35	Number of Exterior Gateway Protocol (EGP) packets	$O(n)$
F36	Number of incomplete packets	$O(n)$
F37	Average Packet size (B)	$O(n)$
F38	Number of BGP update messages (before filtering)	$O(n)$
F39	Number of BGP update messages (after filtering)	$O(n)$
F40	Total size of BGP update messages (before filtering)	$O(n)$
F41	Total size of BGP update messages (after filtering)	$O(n)$

Sliding Time Windows

- Sliding time windows (RIPE Iraq dataset):

POSIXTIME	F1	F2	(...)	F41	LABEL
1692579420	1894	16		314641	0
1692579480	3290	42		469777	0
1692579540	4254	37		620442	0
1692579600	16462	170		2537858	1
1692579660	44204	575		6381473	1

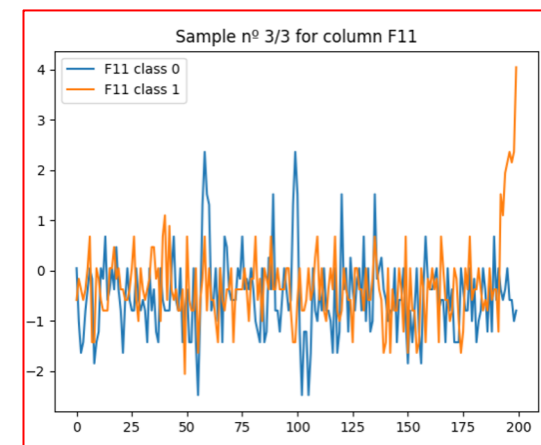
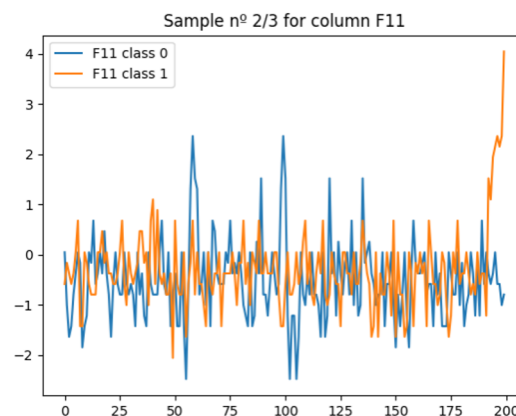
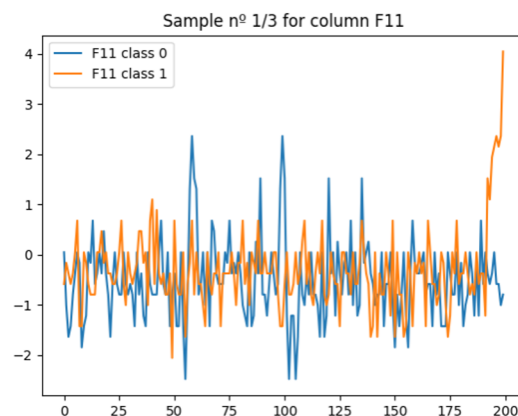
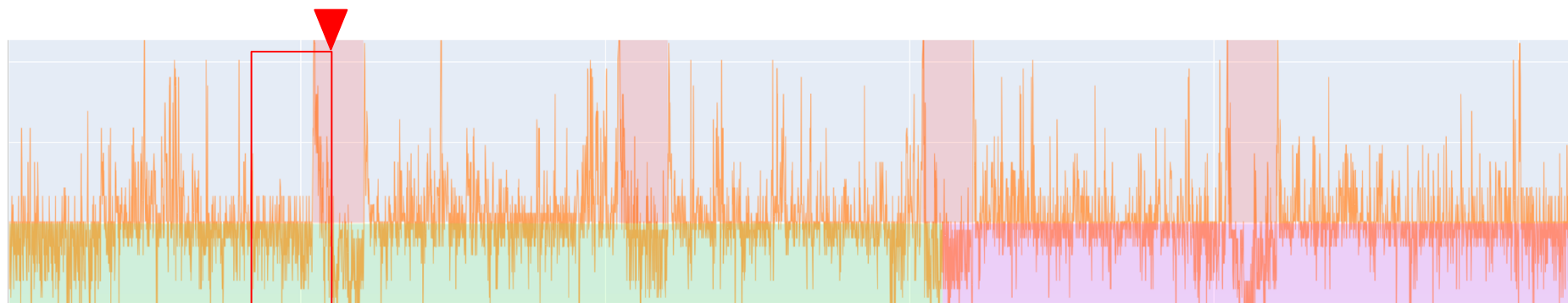


W1

W2

W3

Sliding Time Windows



Data Partitioning

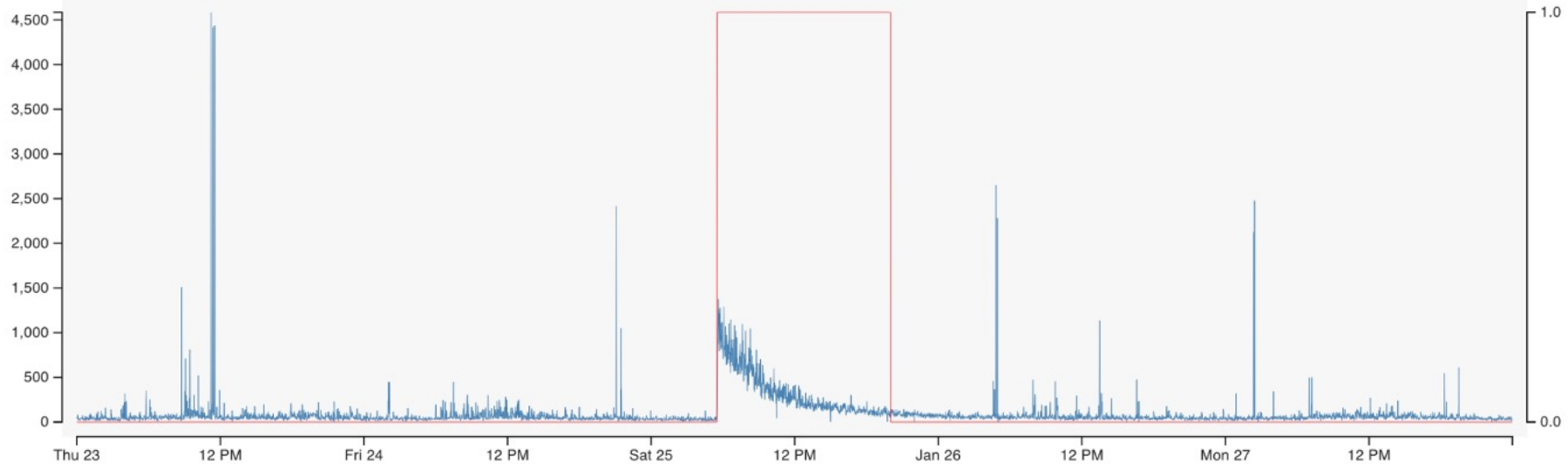
- Data partitioning:
 - Datasets consist of regular and anomalous data points.
 - Anomalous data points:
 - collected during the reported event,
 - Regular data points:
 - collected before and after the outage event.
 - Datasets partitioned to include 60% of anomalous data for training.

Time Series Data Partition

- Partitions based on the entire data

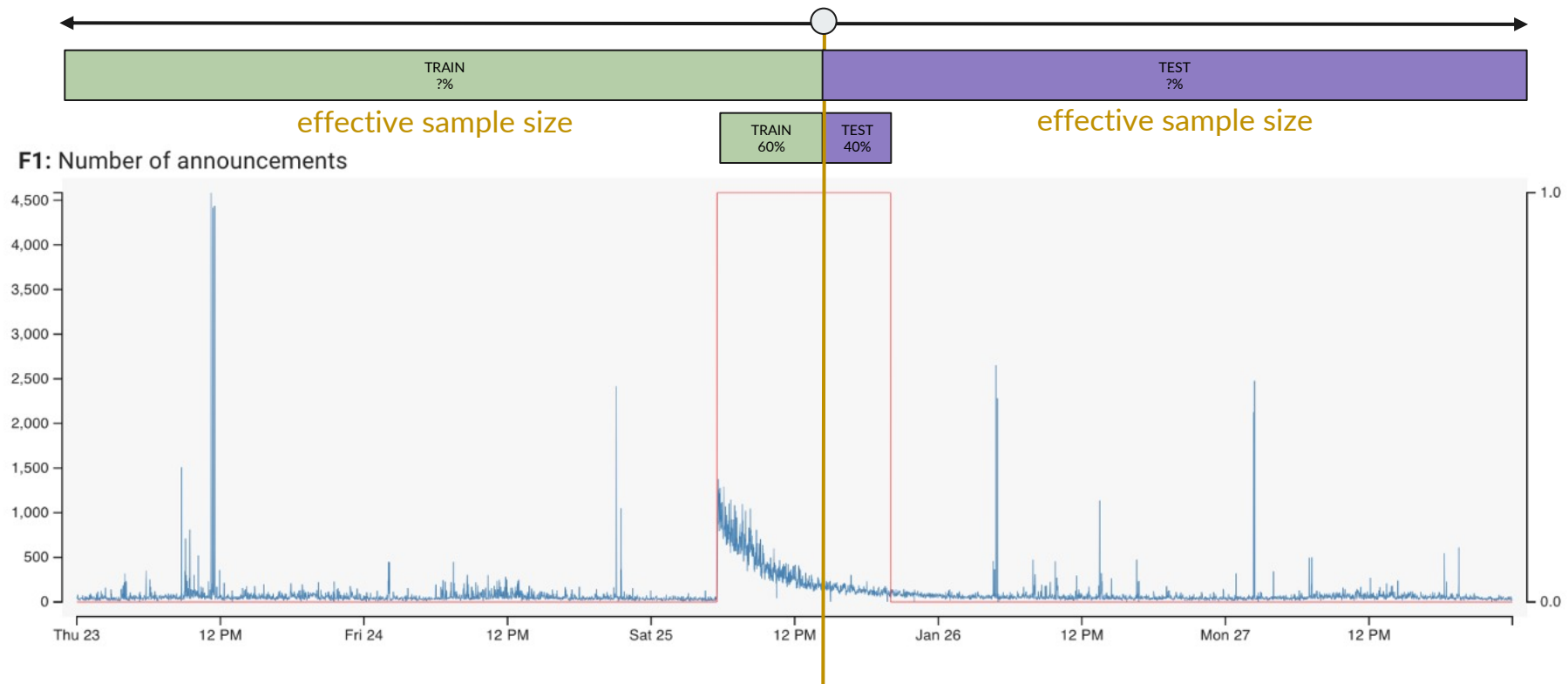


F1: Number of announcements



Time Series Data Partition

- Partitions based on the anomalous data



Selecting Parameters

- **Hyper-parameters:**
 - control the learning process
- **Model parameters:**
 - generated during the training phase, verified using an exhaustive grid search strategy using combinations of hyper-parameters:
 - normalization algorithm: Z-Score
 - feature selection algorithm: extra-trees
 - number of features: 1 to N, where N is the number of features of the dataset
 - length of time window: 1, 10, 20, 30, 40, 50, 100, 150, 200, 250, 300
 - training vs. test ratio: 0.6
 - machine learning model: FCNN, Bi-LSTM
 - number of epochs: 10, 20, 30, 40, 50
 - batch size: 32.

Selecting Hyper-Parameters

- Consider:
 - number of data points, selected features, and length of the time window.
- Performance of models:
 - based on a **fully convolutional neural network** (FCNN) and **Bi-LSTM**.
- **FCNNs** applied to time series data (feature values over time):
 - kernel size = 3; filters = 64
- **Bi-LSTM model**:
 - two Bi-LSTM layers with 128 and 64 blocks each
 - followed by a dense (fully connected) layer with 64 blocks.
- The output layer with sigmoid activation function generates the final binary classification.

FCNN Parameter Grid: Exhaustive Search

L	M	O	P	Q	R	S	T	U
algo	anomaly	epochs	normalization	time_steps	top_n_features	validation	columns	training_time
cnn	0.6	10	fscore	1	1	0.2	F13	30.23664689
cnn	0.6	10	fscore	1	2	0.2	F13, F38	31.86708999
cnn	0.6	10	fscore	1	3	0.2	F13, F38, F40	31.07814646
cnn	0.6	10	fscore	1	4	0.2	F13, F38, F40, F2	30.48660159
cnn	0.6	10	fscore	1	5	0.2	F13, F38, F40, F2, F11	31.68739653
cnn	0.6	10	fscore	1	6	0.2	F13, F38, F40, F2, F11, F10	30.71533418
cnn	0.6	10	fscore	1	7	0.2	F13, F38, F40, F2, F11, F10, F4	30.77034163
cnn	0.6	10	fscore	1	8	0.2	F13, F38, F40, F2, F11, F10, F4, F3	30.7093153
cnn	0.6	10	fscore	1	9	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37	30.96459699
cnn	0.6	10	fscore	1	10	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41	31.48570395
cnn	0.6	10	fscore	1	11	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5	30.90115786
cnn	0.6	10	fscore	1	12	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9	31.44317007
cnn	0.6	10	fscore	1	13	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7	30.08219099
cnn	0.6	10	fscore	1	14	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8	29.96219254
cnn	0.6	10	fscore	1	15	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39	29.68766713
cnn	0.6	10	fscore	1	16	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34	30.35319448
cnn	0.6	10	fscore	1	17	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1	30.16943383
cnn	0.6	10	fscore	1	18	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12	30.00554037
cnn	0.6	10	fscore	1	19	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36	29.98148417
cnn	0.6	10	fscore	1	20	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6	30.72406363
cnn	0.6	10	fscore	1	21	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25	29.85637879
cnn	0.6	10	fscore	1	22	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14	29.78776932
cnn	0.6	10	fscore	1	23	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	29.83635473
cnn	0.6	10	fscore	1	24	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	29.47078919
cnn	0.6	10	fscore	1	25	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	27.6152029
cnn	0.6	10	fscore	1	26	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	28.39402246
cnn	0.6	10	fscore	1	27	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	28.98710966
cnn	0.6	10	fscore	1	28	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	29.24858785
cnn	0.6	10	fscore	1	29	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	29.15906572
cnn	0.6	10	fscore	1	30	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	28.5972259
cnn	0.6	10	fscore	1	31	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	28.84793544

Bi-LSTM Parameter Grid: Exhaustive Search

algo	anomaly	epochs	normalization	time_steps	top_n_features	validation	columns	training_time
bilstm	0.6	50	fscore	200	1	0.2	F13	6179.660032
bilstm	0.6	50	fscore	200	2	0.2	F13, F38	672.9185069
bilstm	0.6	50	fscore	200	3	0.2	F13, F38, F40	881.6231453
bilstm	0.6	50	fscore	200	4	0.2	F13, F38, F40, F2	1438.752977
bilstm	0.6	50	fscore	200	5	0.2	F13, F38, F40, F2, F11	6601.539561
bilstm	0.6	50	fscore	200	6	0.2	F13, F38, F40, F2, F11, F10	6595.58532
bilstm	0.6	50	fscore	200	7	0.2	F13, F38, F40, F2, F11, F10, F4	6635.612075
bilstm	0.6	50	fscore	200	8	0.2	F13, F38, F40, F2, F11, F10, F4, F3	6779.857712
bilstm	0.6	50	fscore	200	9	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37	6745.178365
bilstm	0.6	50	fscore	200	10	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41	6799.976989
bilstm	0.6	50	fscore	200	11	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5	6796.797745
bilstm	0.6	50	fscore	200	12	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9	6759.186007
bilstm	0.6	50	fscore	200	13	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7	6827.209143
bilstm	0.6	50	fscore	200	14	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8	6821.512541
bilstm	0.6	50	fscore	200	15	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39	570.2426243
bilstm	0.6	50	fscore	200	16	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34	6868.504122
bilstm	0.6	50	fscore	200	17	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1	666.4162164
bilstm	0.6	50	fscore	200	18	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12	1780.897617
bilstm	0.6	50	fscore	200	19	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36	791.9169838
bilstm	0.6	50	fscore	200	20	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6	6988.384876
bilstm	0.6	50	fscore	200	21	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25	6967.352288
bilstm	0.6	50	fscore	200	22	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14	6960.486874
bilstm	0.6	50	fscore	200	23	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	7015.823389
bilstm	0.6	50	fscore	300	32	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	5600.985467
bilstm	0.6	50	fscore	300	33	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	5535.149159
bilstm	0.6	50	fscore	300	34	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	5628.222851
bilstm	0.6	50	fscore	300	35	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	5645.803564
bilstm	0.6	50	fscore	300	36	0.2	F13, F38, F40, F2, F11, F10, F4, F3, F37, F41, F5, F9, F7, F8, F39, F34, F1, F12, F36, F6, F25, F14, F15	5739.124215

Performance Evaluation and Model Selection

- We measure the model's ability to predict the labels based on:
 - Accuracy and F-Score
 - confusion matrix:
true positive (TP), true negative (TN), false positive (FP), and false negative (FN)
 - training time.
- The model training and test steps were performed for each combination of hyperparameter in the exhaustive grid search.
- The best model is chosen based on:
 - the highest accuracy
 - the shortest training time.
- Training time: critical in cybersecurity applications where online models are trained using network real-time data.

Performance Metrics

- Training time
- Accuracy:
 - $(TP + TN) / (TP + TN + FP + FN)$
- F-Score signifies harmonic mean between precision and sensitivity (recall):
 - $2 \times (\text{precision} \times \text{sensitivity}) / (\text{precision} + \text{sensitivity})$

where:

- Precision
 - $TP / (TP + FP)$
- Sensitivity:
 - $TP / (TP + FN)$
- Confusion matrix: TP, FP, TN, FN

Performance Metrics: Definitions

- **Accuracy**

Accuracy is the fraction of the models predictions that were correct. Formally, accuracy is $\text{Number of correct predictions} / \text{Total number of predictions}$.

- **F1-score**

The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

- **Precision**

The precision is the ability of the classifier not to label as positive a sample that is negative.

- **Recall**

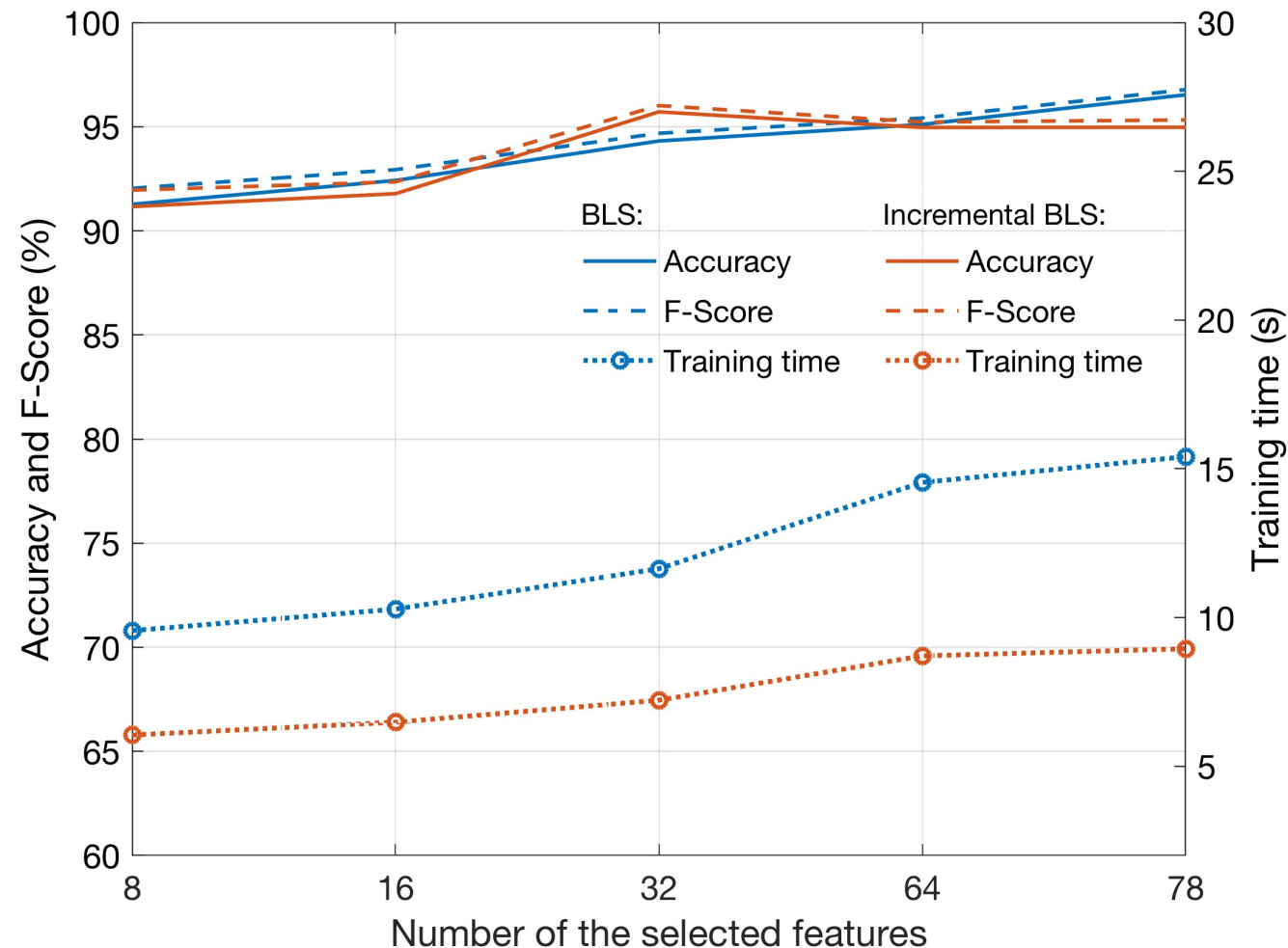
The recall is the ability of the classifier to find all the positive samples.

Roadmap

- Introduction
- Data collections
- Feature selection
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- **Performance evaluation**
- Conclusions and references

Performance:

BLS and Incremental BLS, CIC 2017



Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
CNN	No refinement	RIPE	18.79	55.33	70.96	57.04	93.85
		RouteViews	18.66	57.67	72.96	58.04	98.23
	k-means	RIPE	19.31	55.31	71.00	56.99	94.25
		RouteViews	18.88	57.06	72.32	57.82	96.52
	Isolation forest	RIPE	19.20	55.29	70.96	57.00	94.00
		RouteViews	18.80	57.12	72.44	57.83	96.93

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
GRU ₄	No refinement	RIPE	13.99	75.23	80.34	74.84	86.48
LSTM ₄		RouteViews	18.95	55.42	70.72	57.20	92.60
GRU ₄	k-means	RIPE	14.44	75.44	79.73	76.63	83.10
GRU ₂		RouteViews	13.44	62.30	69.61	65.47	74.31
LSTM ₂	Isolation forest	RIPE	12.63	75.36	79.73	76.41	83.35
LSTM ₃		RouteViews	13.77	60.00	69.06	62.75	76.80

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
Bi-GRU ₄	No refinement	RIPE	20.59	78.49	81.92	80.10	83.83
Bi-GRU ₃		RouteViews	21.89	62.50	69.70	65.73	74.18
Bi-GRU ₃	k-means	RIPE	20.27	77.76	82.05	77.30	87.43
		RouteViews	20.14	63.36	72.15	64.61	81.69
Bi-GRU ₄	Isolation forest	RIPE	23.73	84.27	86.90	84.23	89.75
Bi-GRU ₃		RouteViews	20.23	64.74	72.19	66.67	78.70

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
RBF-BLS	No refinement	RIPE	3.98	55.70	70.75	57.41	92.18
		RouteViews	2.60	54.74	69.99	56.95	90.78
	Isolation forest	RIPE	2.20	55.73	70.77	57.42	92.20
		RouteViews	3.97	54.61	69.81	56.91	90.28

Best Performance: WestRock Ransomware

Model Incremental		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
RBF-BLS	No refinement	RIPE	1.71	58.20	73.55	58.18	99.98
CEBLS		RouteViews	23.33	57.89	73.31	58.05	99.48
RBF-BLS	Isolation forest	RIPE	33.28	58.20	73.54	58.16	99.98
		RouteViews	7.01	58.15	73.52	58.16	99.93

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
VFBLS	No refinement	RIPE	7.31	55.15	70.18	57.19	90.80
		RouteViews	7.99	54.75	69.92	56.99	90.45
	Isolation forest	RIPE	6.18	54.74	69.81	57.00	90.05
		RouteViews	5.67	54.23	69.41	56.76	89.33

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
VCFBLS	No refinement	RIPE	4.14	55.33	70.31	57.30	90.95
		RouteViews	4.62	54.68	69.73	56.99	89.80
	Isolation forest	RIPE	6.56	54.72	69.86	56.98	90.27
		RouteViews	4.66	54.43	69.55	56.87	89.53

Best Performance: WestRock Ransomware

Model Incremental		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
VFBLS	No refinement	RIPE	6.77	58.17	73.54	58.16	100
		RouteViews	6.82	58.18	73.55	58.16	100
	Isolation forest	RIPE	11.60	58.27	73.55	58.23	99.80
		RouteViews	7.62	58.20	73.55	58.18	99.98

Best Performance: WestRock Ransomware

Model Incremental		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
VCFBLS	No refinement	RIPE	12.04	58.23	73.57	58.19	99.98
		RouteViews	9.08	58.30	73.57	58.25	99.85
	Isolation forest	RIPE	11.27	58.15	73.53	58.14	99.98
		RouteViews	10.40	58.20	73.56	58.17	100

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
XGBoost	No refinement	RIPE	0.54	60.44	73.38	60.26	93.80
		RouteViews	0.27	55.83	70.94	57.44	92.73
	Isolation forest	RIPE	0.52	59.84	73.05	59.88	93.62
		RouteViews	0.38	55.58	70.42	57.46	90.93

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
LightGBM	No refinement	RIPE	0.05	58.37	72.20	59.01	92.98
		RouteViews	0.06	57.50	72.16	58.27	94.73
	k-means	RIPE	0.14	58.11	71.29	59.25	89.48
		RouteViews	0.07	57.56	72.53	58.13	96.42
	Isolation forest	RIPE	0.10	57.66	71.42	58.77	91.02
		RouteViews	0.05	57.72	72.81	58.14	97.38

Best Performance: WestRock Ransomware

Model		Collection site	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)
CatBoost	No refinement	RIPE	0.33	55.60	71.36	57.09	95.15
		RouteViews	0.31	58.17	73.53	58.16	99.95
	Isolation forest	RIPE	0.32	55.58	71.34	57.07	95.12
		RouteViews	0.48	58.24	73.53	58.22	99.78

Performance Results

TABLE II
PERFORMANCE RESULTS: BEST-PERFORMING HYPER-PARAMETERS AND PERFORMANCE METRICS

Collection Site	Dataset	Algorithm	Hyper-Parameters			Performance Metrics						
			Size of time window	Number of epochs	Number of features	Accuracy %	F-Score %	TP	TN	FP	FN	Training time (s)
IODA	Iraq	FCNN	10	10	3	1	1	154	1,008	0	0	107.23
		Bi-LSTM	1	50	4	1	1	154	1,008	0	0	48.53
	Gaza Strip	FCNN	20	20	2	99.2	97.6	272	1,511	6	7	114.10
		Bi-LSTM	150	50	1	99.4	98.3	274	1,513	4	5	2,401.23
	Ukraine	FCNN	10	10	2	1	1	55	288	0	0	96.39
		Bi-LSTM	20	20	2	1	1	55	288	0	0	184.04
RIPE	Iraq	FCNN	200	20	13	99.6	98.6	373	2,637	7	3	179.12
		Bi-LSTM	250	30	37	99.9	99.7	374	2,644	0	2	11,998.88
	Gaza Strip	FCNN	100	40	20	84.7	4.0	29	7,578	15	1,359	806.83
		Bi-LSTM	300	20	2	84.8	7.2	53	7,571	22	1,335	9,557.98
	Ukraine	FCNN	200	40	9	89.5	68.7	197	1,339	105	74	363.20
		Bi-LSTM	50	40	6	85.2	12.4	18	1,444	0	253	901.01

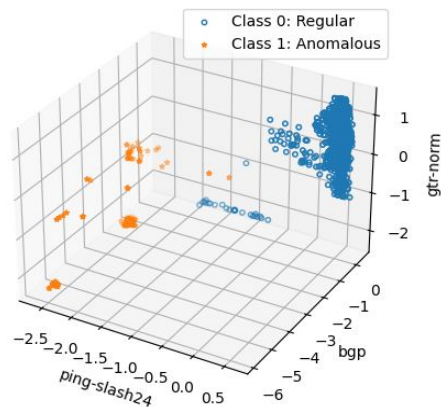
Performance Results: IODA

- Models trained using IODA datasets exhibit superior performance with 100% accuracy and F-Score for several combinations of hyper-parameters.
- The models detect outages with high accuracy, even when using one feature and no sliding time window.
- The features are highly correlated to the anomaly periods.
- The IODA measurements were **specifically designed to detect outages**, hence, the unusually high accuracy of models based on the IODA datasets.

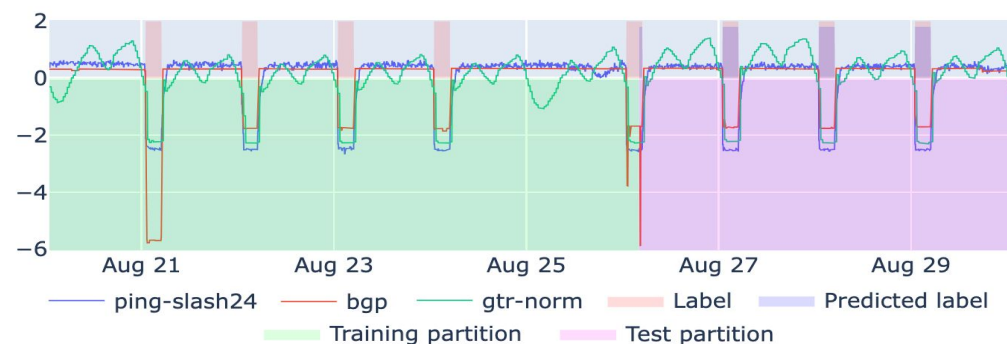
Performance Results: RIPE

- Detecting outages using RIPE datasets proves more challenging.
- No significant spatial separation is observed for the RIPE Iraq dataset.
- No visible correlation between the feature values and the anomaly periods.
- The 41 features extracted from the RIPE datasets were successfully used for the classification of network anomalies such as worms, viruses, and ransomware.
- The best accuracies based on RIPE datasets:
 - Iraq: 99.9% (Bi-LSTM)
 - The Gaza Strip: 84.8% (Bi-LSTM)
 - Ukraine: 89.5% (FCNN).

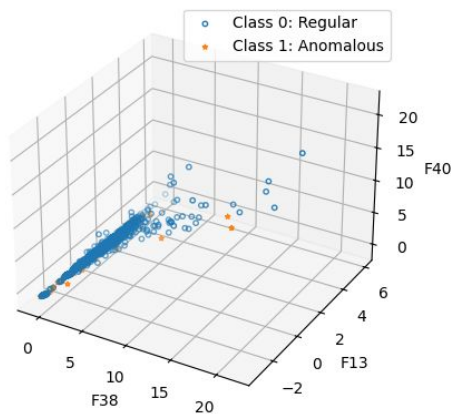
Performance Results



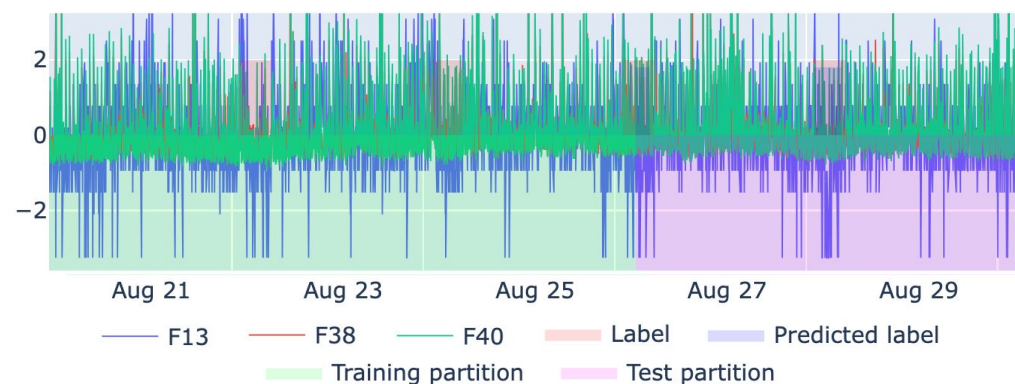
(a)



(b)



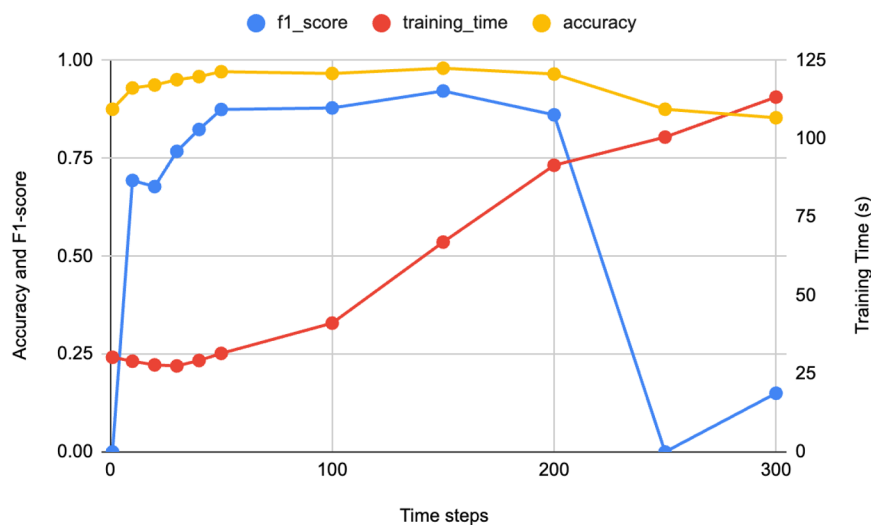
(c)



(d)

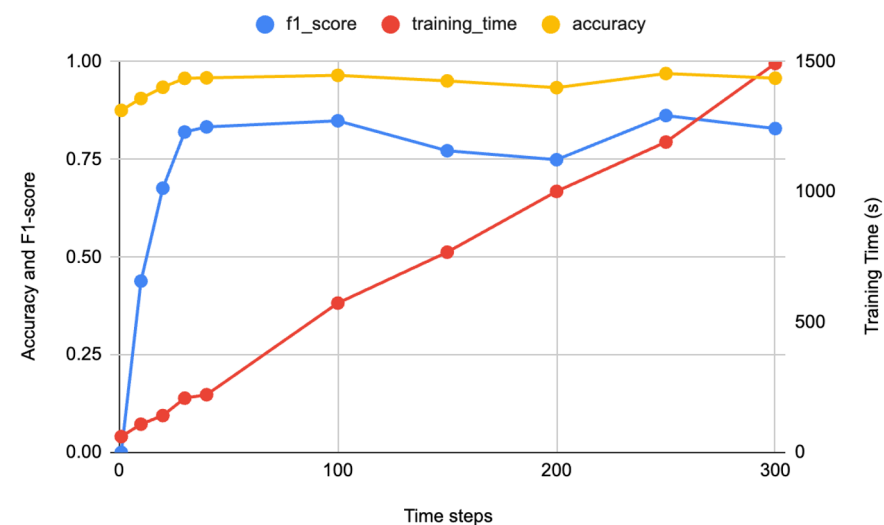
3D scatter plots of (a) Iraq IODA and (c) RIPE datasets display the spatial separation between regular and anomalous data points. The three most important features of the training and test partitions and the periods labeled and then predicted as anomalous from the (b) Iraq IODA and (d) RIPE datasets.

Performance Results



FCNN

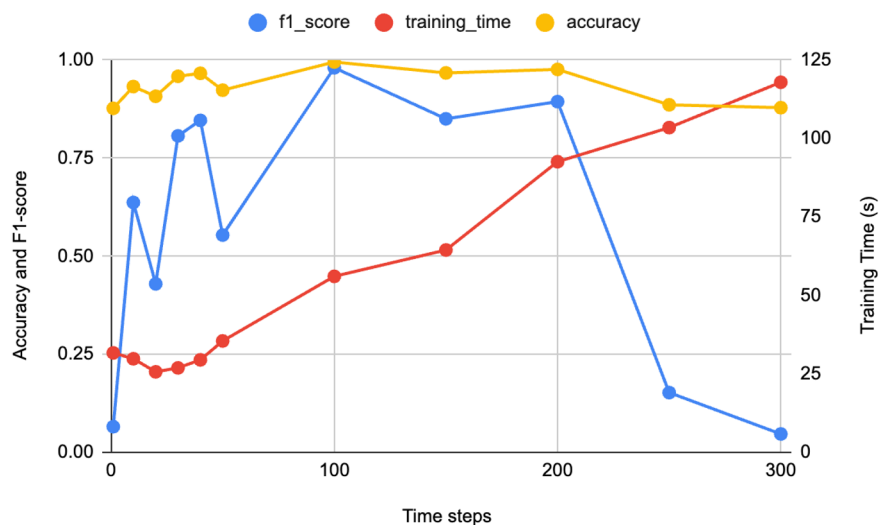
Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 1
Normalization = F-Score
RIPE RRC00 Iraq dataset



Bi-LSTM

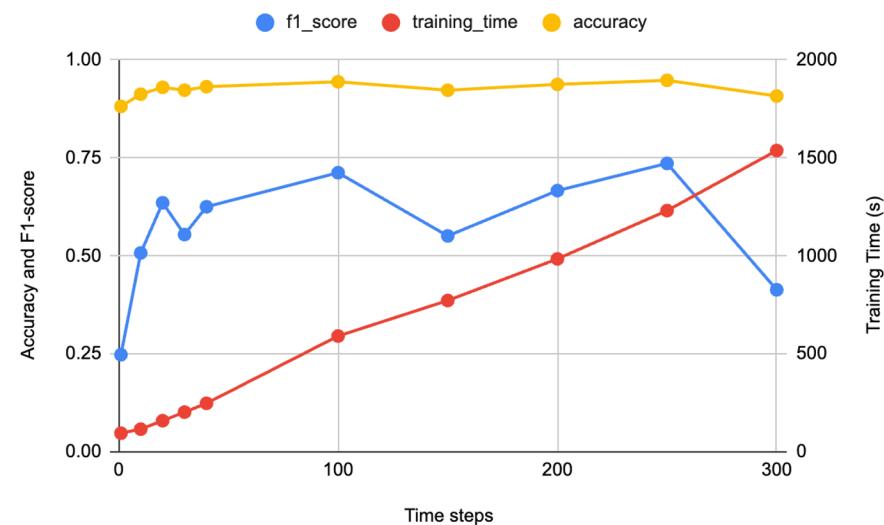
Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 1
Normalization = F-Score
RIPE RRC00 Iraq dataset

Performance Results



FCNN

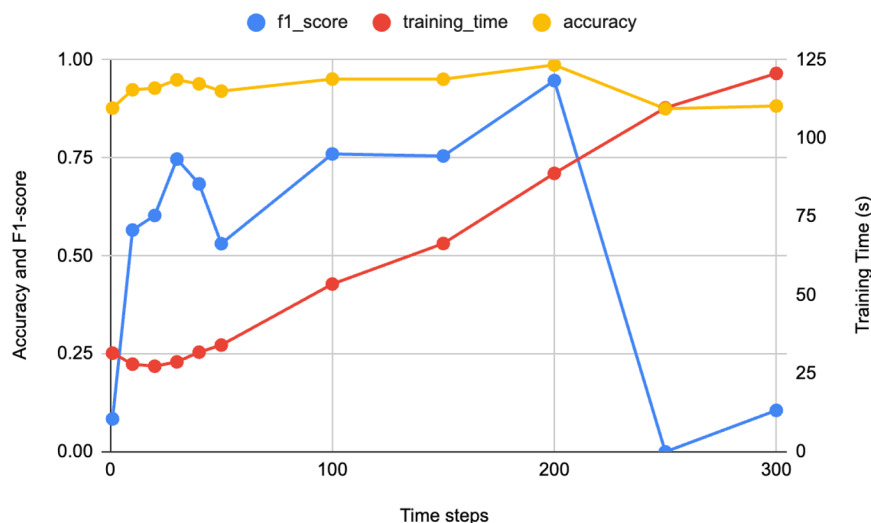
Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 5
Normalization = F-Score
RIPE RRC00 Iraq dataset



Bi-LSTM

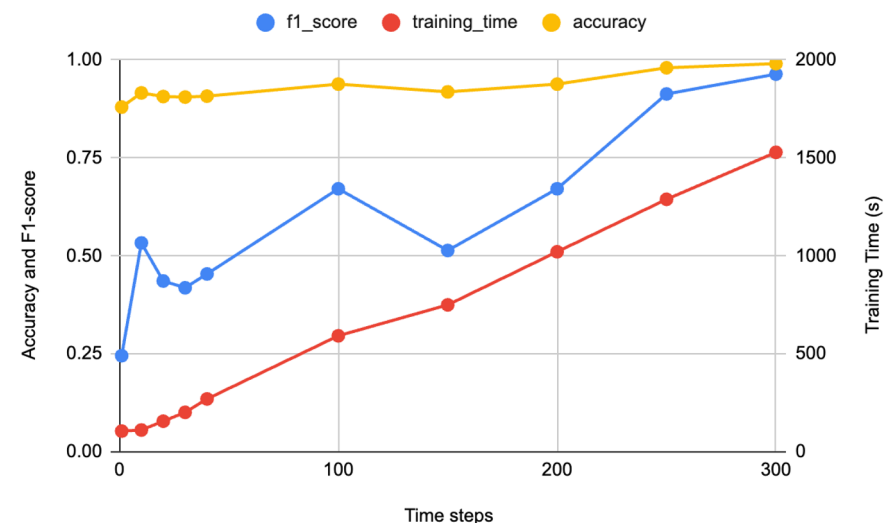
Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 5
Normalization = F-Score
RIPE RRC00 Iraq dataset

Performance Results



FCNN

Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 10
Normalization = F-Score
RIPE RRC00 Iraq dataset



Bi-LSTM

Anomalous Ratio = 0.6, Epochs = 10,
Top N-features = 10
Normalization = F-Score
RIPE RRC00 Iraq dataset

Performance Results



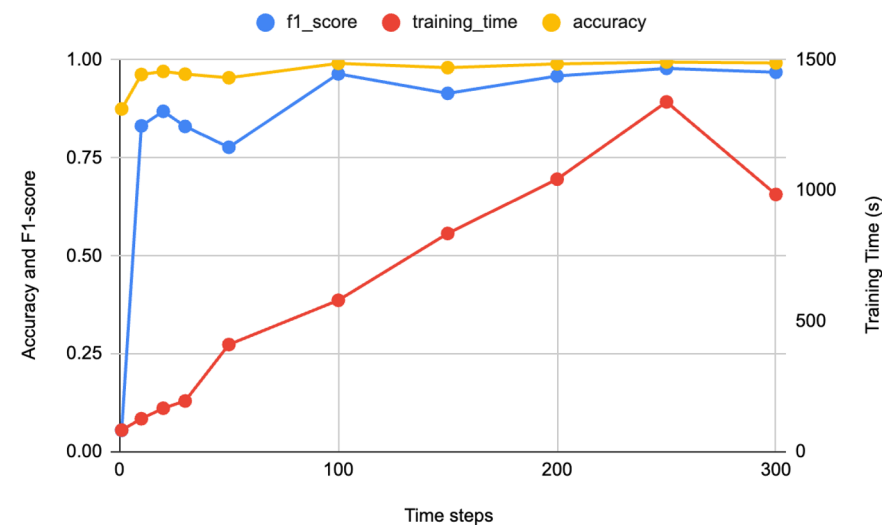
FCNN

Anomalous Ratio = 0.6, Epochs = 10,

Top N-features = 41

Normalization = F-Score

RIPE RRC00 Iraq dataset



Bi-LSTM

Anomalous Ratio = 0.6, Epochs = 10,

Top N-features = 41

Normalization = F-Score

RIPE RRC00 Iraq dataset

Roadmap

- Introduction
- Data collections
- Machine learning models:
 - Deep learning networks
 - Broad learning system
 - Gradient boosting decision trees
- Experimental procedure
- Performance evaluation
- Conclusions and references

Conclusions

- We evaluated performance of:
 - CNNs
 - RNNs: LSTM, Bi-LSTM, GRU, and Bi-GRU deep recurrent neural networks with a variable number of hidden layers
 - BLS models with and without incremental learning:
 - radial basis function
 - cascades of mapped features and enhancement nodes
 - integrated extra-trees for feature selection (VFBLs and VCFBLs)
 - GBDT: XGBoost, LightGBM, CatBoost

References: Tools

- Python: <https://pypi.org>
Pandas: <https://pandas.pydata.org/>
- PyTorch
<https://pytorch.org/docs/stable/nn.html>
- zebra-dump-parser:
<https://github.com/rfc1036/zebra-dump-parser>
- BGP C# tool:
http://www.sfu.ca/~ljilja/cnl/projects/BGP_datasets/index.html
- IEEE DataPort
Border Gateway Protocol (BGP) datasets:
 - <https://ieee-dataport.org/open-access/border-gateway-protocol-bgp-routing-records-reseaux-ip-europeens-ripe-and-bcnet>
 - <https://ieee-dataport.org/open-access/border-gateway-protocol-bgp-routing-records-route-views>
- BLS: Broadlearning: <http://www.broadlearning.ai/>

References: Intrusion Detection

- J. P. A. Maranhão, J. P. C. L. da Costa, E. P. de Freitas, E. Javidi, and R. T. de Sousa, Jr., “Noise-robust multilayer perceptron architecture for distributed denial of service attack detection,” *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 402–406, Feb. 2021.
- P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” *IEEE Commun. Surveys Tut.*, vol. 21, no. 1, pp. 686–728, First quarter 2019.
- A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Commun. Surv. Tut.*, vol. 18, no. 2, pp. 1153–1176, 2016.
- M. C. Libicki, L. Ablon, and T. Webb, *The Defenders Dilemma: Charting a Course Toward Cybersecurity*. Santa Monica, CA, USA: RAND Corporation, 2015.
- 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.

References: Deep Learning

- S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Oct. 1997.
- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *Computing Research Repository (CoRR)*, abs/1207.0580, pp. 1–18, Jul. 2012.
- K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translations,” in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1724–1734.
- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: The MIT Press, 2016.
- 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.

References: BLS and GBDT

- C. L. P. Chen and Z. Liu, “Broad learning system: an effective and efficient incremental learning system without the need for deep architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- C. L. P. Chen, Z. Liu, and S. Feng, “Universal approximation capability of broad learning system and its structural variations,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.
- T. Chen and C. Guestrin, “XGBoost: a scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
- G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, “LightGBM: a highly efficient gradient boosting decision tree,” in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, 3146–3154.
- L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” in *Proc. Int. Conf. Neural Inform. Process. Syst.*, Montreal, Quebec, Canada, Dec. 2018, 6639–6649.

Publications: <http://www.sfu.ca/~ljilja>

Journal publication:

- Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, “Machine learning for detecting the WestRock ransomware attack using BGP routing records,” *IEEE Communications Magazine*, vol. 61, no. 3, pp. 20–26, Mar. 2023.
- Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, “Machine learning for detecting anomalies and intrusions in communication networks,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2254–2264, July 2021.

Book chapters:

- Q. Ding, Z. Li, S. Haeri, and Lj. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks: datasets and feature selection algorithms” in *Cyber Threat Intelligence*, M. Conti, A. Dehghantanha, and T. Dargahi, Eds., Berlin: Springer, pp. 47–70, 2018.
- 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, pp. 71–92, 2018.

Publications: <http://www.sfu.ca/~ljilja>

Conference publications:

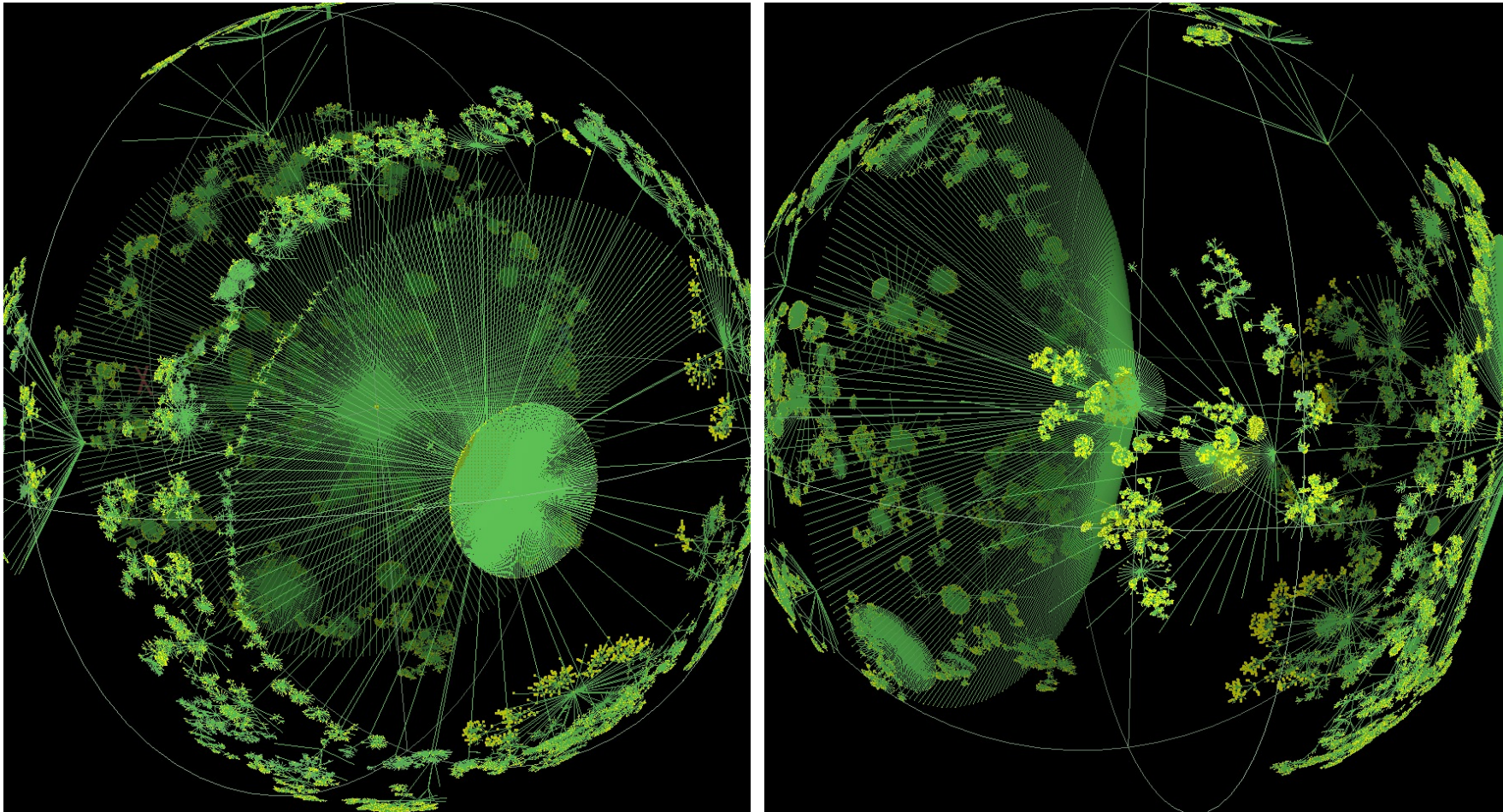
- L. F. Oliveira, R. Ballantyne, J. Souza, and Lj. Trajkovic, "Internet outages during times of conflict," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Kuching, Sarawak, Malaysia, Oct. 2024.
- H. Takhar, L. F. Oliveira, and Lj. Trajković, "Case study: understanding Internet anomalies," in *Proc. IEEE Int. Symp. Circuits and Systems*, Singapore, May 2024.
- Z. Li and Lj. Trajković, "CyberDefense: tool for detecting network anomalies and intrusions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Honolulu, HI, USA, Oct. 2023.
- H. Takhar and Lj. Trajković, "BGP feature properties and classification of Internet worms and ransomware attacks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Honolulu, HI, USA, Oct. 2023.
- T. Sharma, K. Patni, Z. Li, and Lj. Trajković, "Deep echo state networks for detecting Internet worm and ransomware attacks" in *Proc. IEEE Int. Symp. Circuits and Systems*, Monterey, CA, USA, May 2023.
- Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, "Classifying denial of service attacks using fast machine learning algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Melbourne, Australia, Oct. 2021, pp. 1221-1226 (virtual).
- K. Bekshentayeva and Lj. Trajkovic, "Detection of denial of service attacks using echo state networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Melbourne, Australia, Oct. 2021, pp. 1227-1232 (virtual).

Publications: <http://www.sfu.ca/~ljilja>

Conference publications:

- Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, “Detecting Internet worms, ransomware, and blackouts using recurrent neural networks,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Toronto, Canada, Oct. 2020, pp. 2165–2172 (virtual).
- A. L. Gonzalez Rios, Z. Li, K. Bekshentayeva, and Lj. Trajković, “Detection of denial of service attacks in communication networks,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Seville, Spain, Oct. 2020 (virtual).
- Z. Li, A. L. Gonzalez Rios, G. Xu, and Lj. Trajković, “Machine learning techniques for classifying network anomalies and intrusions,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Sapporo, Japan, May 2019 (virtual).
- A. L. Gonzalez Rios, Z. Li, G. Xu, A. Dias Alonso, and Lj. Trajković, “Detecting network anomalies and intrusions in communication networks,” in *Proc. 23rd IEEE International Conference on Intelligent Engineering Systems 2019*, Gödöllő, Hungary, Apr. 2019, pp. 29–34.
- Z. Li, P. Batta, and Lj. Trajković, “Comparison of machine learning algorithms for detection of network intrusions,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Miyazaki, Japan, Oct. 2018, pp. 4248–4253.
- P. Batta, M. Singh, Z. Li, Q. Ding, and Lj. Trajković, “Evaluation of support vector machine kernels for detecting network anomalies,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Florence, Italy, May 2018, pp. 1–4.
- Q. Ding, Z. Li, P. Batta, and Lj. Trajković, “Detecting BGP anomalies using machine learning techniques,” in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, Hungary, Oct. 2016, pp. 3352–3355.

Ihr: 535,102 nodes and 601,678 links



<http://www.caida.org/home/>