Classifying Denial of Service Attacks Using Fast Machine Learning Algorithms

Zhida Li, Ana Laura Gonzalez Rios, and Ljiljana Trajković

Communication Networks Laboratory

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

School of Engineering Science

Simon Fraser University, Vancouver
British Columbia, Canada

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Introduction

- Denial of service attacks are harmful cyberattacks that diminish Internet resources and services
- Detecting these cyberattacks is a topic of great interest in cybersecurity
- Denial of service (DoS) attacks: performed from a single system
- Distributed DoS (DDoS) attacks: executed from multiple systems
- Classified as: floods, fragmentation, Transport Control Protocol (TCP) state exhaustion, and application-layer attacks
- Datasets capturing DoS and DDoS attacks have been synthetically generated by the Canadian Institute for Cybersecurity (CIC)

Introduction

- Detection techniques for DoS and DDoS attacks include: activity profiling, change-point detection, wavelet analysis, and machine learning algorithms
- Machine learning algorithms:
  - Support vector machine: SVM
  - Deep neural networks:
    - Convolutional neural networks (CNNs)
    - Recurrent neural networks (RNNs)
    - Autoencoders
    - Multilayer perceptrons
  - Broad learning system: BLS and its extensions
  - Gradient boosting decision trees (GBDT)

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Machine learning algorithms

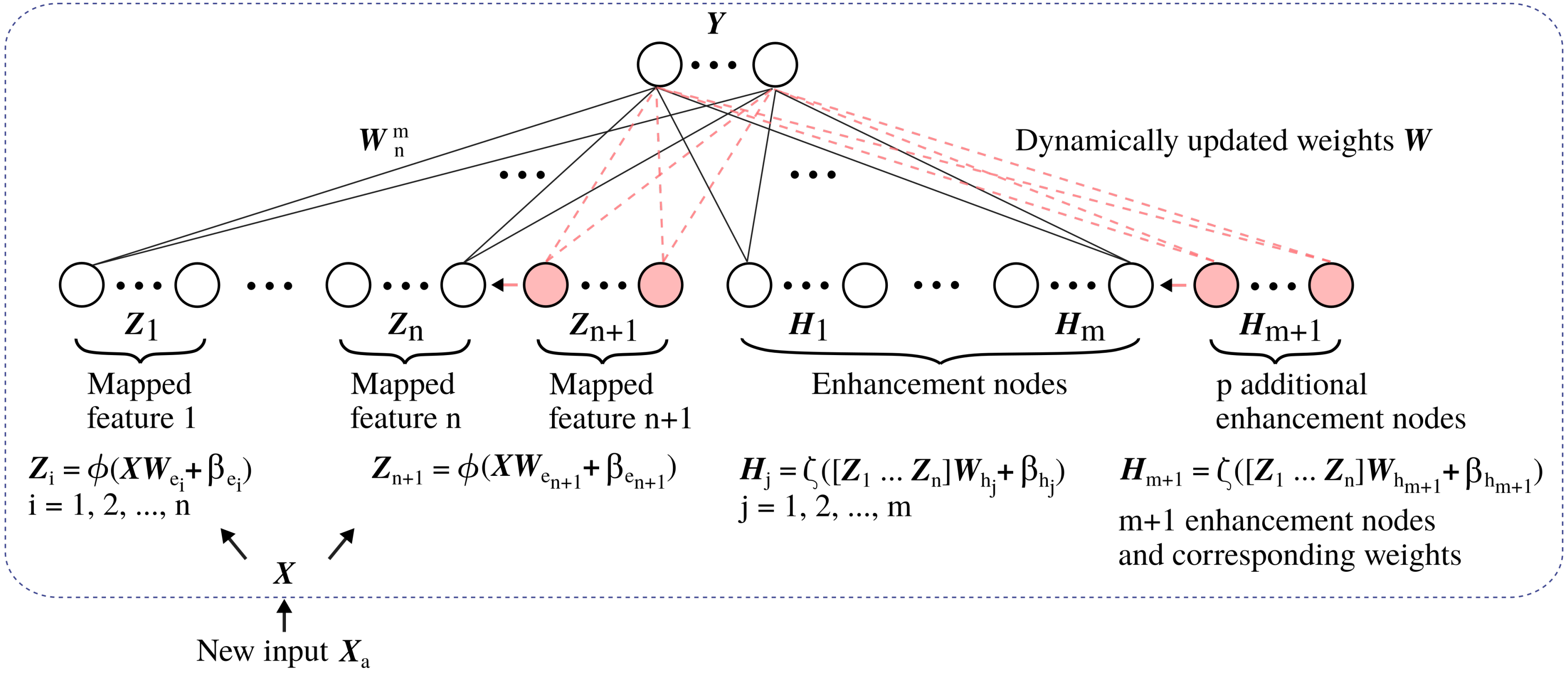
- Detection of DoS and DDoS attacks: require updating or retraining generated models to capture deviations from regular network activities
- Training time:
  - important for the decision-making process at the onset of anomalies when preventing cyberattacks on servers and avoiding DoS to legitimate users
- Fast training machine learning algorithms:
  - BLS:
    - a single layer feed-forward neural network
    - employs pseudo-inverse rather than back-propagation
  - GBDT:
    - an ensemble of decision trees
    - employs functional gradient descent

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Broad learning system

- Broad learning system (BLS) algorithm with increments of mapped features, enhancement nodes, and/or new input data:



Original BLS

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

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

where:

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

Original BLS

- Modified to include additional mapped features $\boldsymbol{Z}_{n+1}$, enhancement nodes $\boldsymbol{H}_{m+1}$, and/or input nodes $\boldsymbol{X}_a$
- Moore-Penrose pseudo inverse of matrix $\boldsymbol{A}_x$ is computed to calculate the weights of the output:

$$\boldsymbol{W}_n^m = [\boldsymbol{A}_n^m]^+\boldsymbol{Y}$$

- During the training process, data labels are deduced using the calculated weights $\boldsymbol{W}_n^m$, mapped features $\boldsymbol{Z}_n$, and enhancement nodes $\boldsymbol{H}_m$:

$$\begin{aligned}\boldsymbol{Y} &= \boldsymbol{A}_n^m\boldsymbol{W}_n^m \\ &= [\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_n|\boldsymbol{H}_1, \dots, \boldsymbol{H}_m]\boldsymbol{W}_n^m\end{aligned}$$

RBF-BLS extension

- The RBF function is implemented using Gaussian kernel:

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

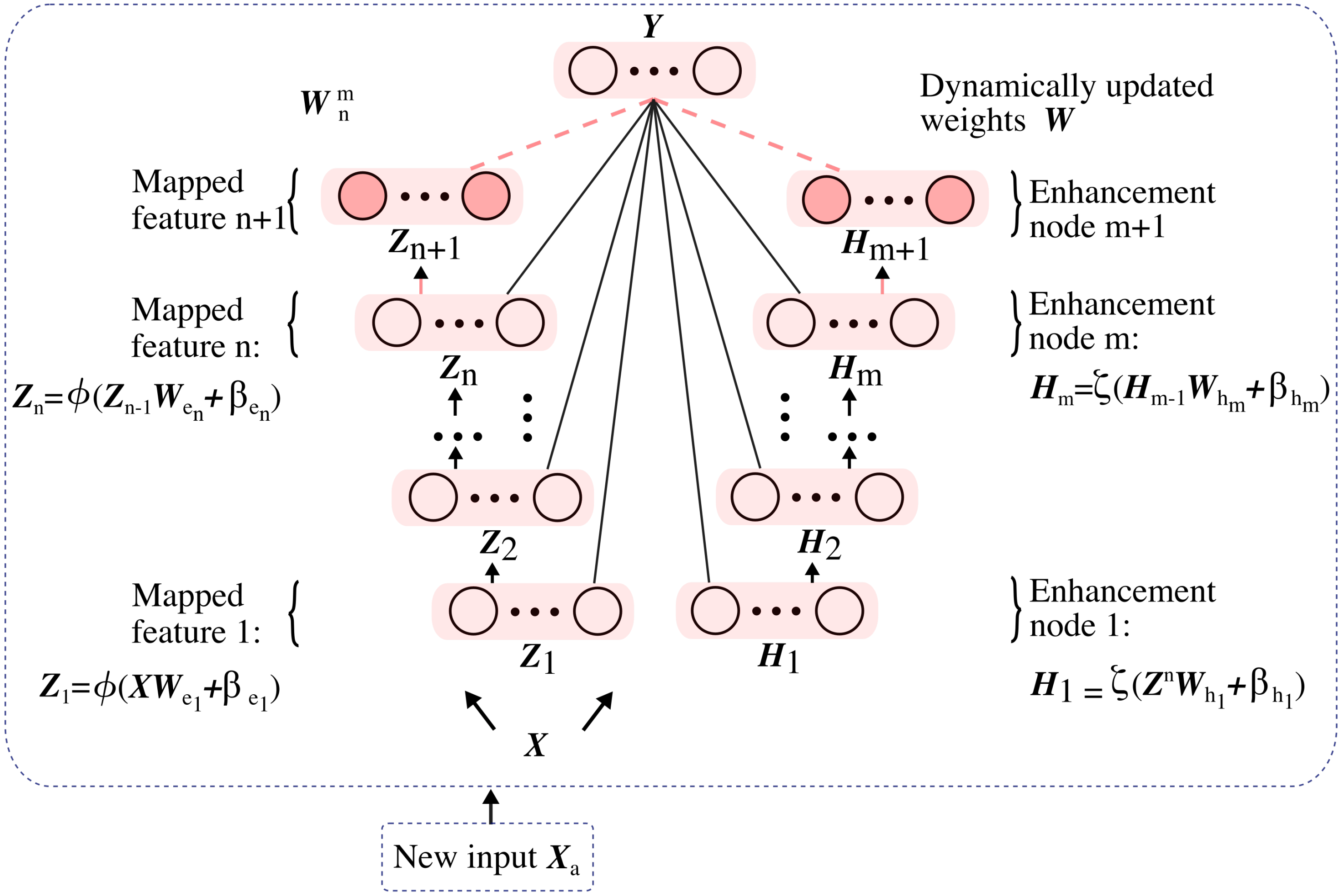
- Weight vectors of the output $\boldsymbol{HW}$ are deduced from:

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

where:

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

Cascades with incremental learning

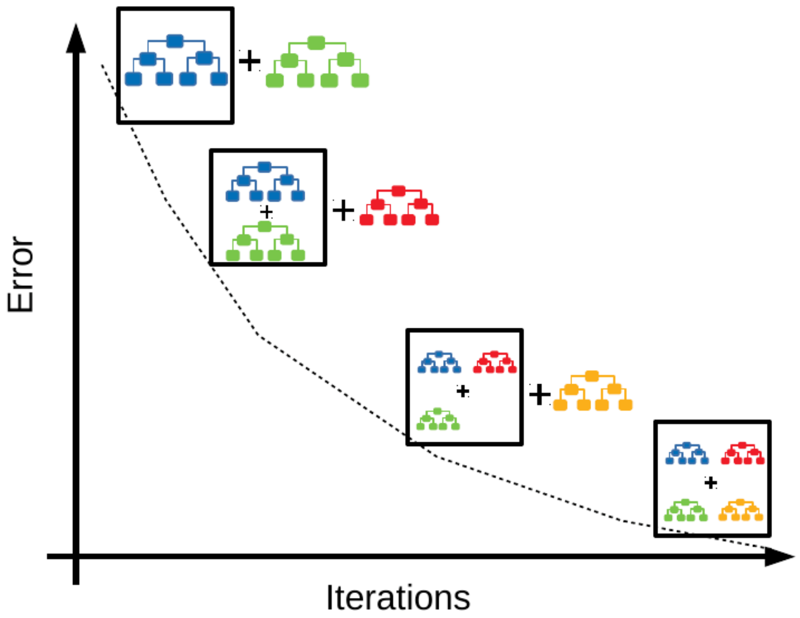


Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Gradient boosting machines

- Gradient boosting machines (GBMs): boosting algorithms that employ functional gradient descent to minimize the loss function
- GBDT: GBM variant that employs decision trees as estimators



Generating a gradient boosting model

https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea

Gradient boosting decision trees

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

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

where:

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

GBDT: XGBoost

- The 2nd order Taylor series approximates the objective function:

$$\mathcal{L}^{(k)} \simeq \sum_{i=1}^{N} \left[l\left(y_i - \hat{y}_i^{(k-1)}\right) + g_i f_k(\boldsymbol{x}_i) + \frac{1}{2} h_i f_k^2(\boldsymbol{x}_i) \right] + \Omega(f_k),$$

where g_i and h_i are the known terms and $l(\cdot)$ is the constant term

- For a known tree structure $q(\boldsymbol{X})$, I_t is a set containing the indices of data points in leaf t
- Setting the derivative of the objective function approximation to zero gives the optimal weight ω_t^* for leaf t:

$$\omega_t^* = -\frac{\sum_{i \in I_t} g_i}{\sum_{i \in I_t} h_i + \lambda}$$

GBDT: XGBoost

- Optimal solution of the objective function:

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

- This optimal value is used to evaluate the quality of a tree structure $q(\boldsymbol{X})$
- 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:

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

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 $\boldsymbol{x}_i$

GBDT: LightGBM

- The sampling ratios a and b are used to calculate the normalization coefficient $\frac{1-a}{b}$
- Subsets of $\boldsymbol{A}(\boldsymbol{B})$:
  - $\boldsymbol{A}_l(\boldsymbol{B}_l)$: left child nodes
  - $\boldsymbol{A}_r(\boldsymbol{B}_r)$: right child nodes

GBDT: CatBoost

- CatBoost is introduced to deal with categorical features
- It employs the ordered boosting algorithm and offers an effective approach when compared to XGBoost and LightGBM
- Target statistic was used to convert categorical features 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 with target statistic and ordered boosting, respectively

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Canadian Institute for Cybersecurity datasets

CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019:

- Testbed used to create the publicly available dataset that includes multiple types of recent cyber attacks
- Dataset features: extracted from collected TCP and UDP network flows with a network traffic flow analyzer
- Each dataset: over 80 features including destination IP and port, protocol type, flow duration, and maximum/minimum packet size
- Network traffic collected:
  - Monday, 03.07.2017 to Friday, 07.07.2017
  - Wednesday, 14.02.2018 to Friday, 02.03.2018
  - Saturday, 03.11.2018 and Saturday, 01.12.2018

CIC datasets: DoS and DDoS attacks

- Application-layer DoS and TCP/UDP DDoS attacks

Dataset	Attack	Number of Data Points
	GoldenEye	10,293
CCIDS2017	Hulk	230,124
July 05, 2017	SlowHTTPTest	5,499
	Slowloris	5,796
CSE-CIC-IDS2018	GoldenEye	41,508
February 15, 2018	Slowloris	10,990
CICDDoS2019	Domain Name System	5,071,011
December 01, 2018	Lightweight Directory Access Protocol	2,179,930
	Network Time Protocol	1,202,642

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

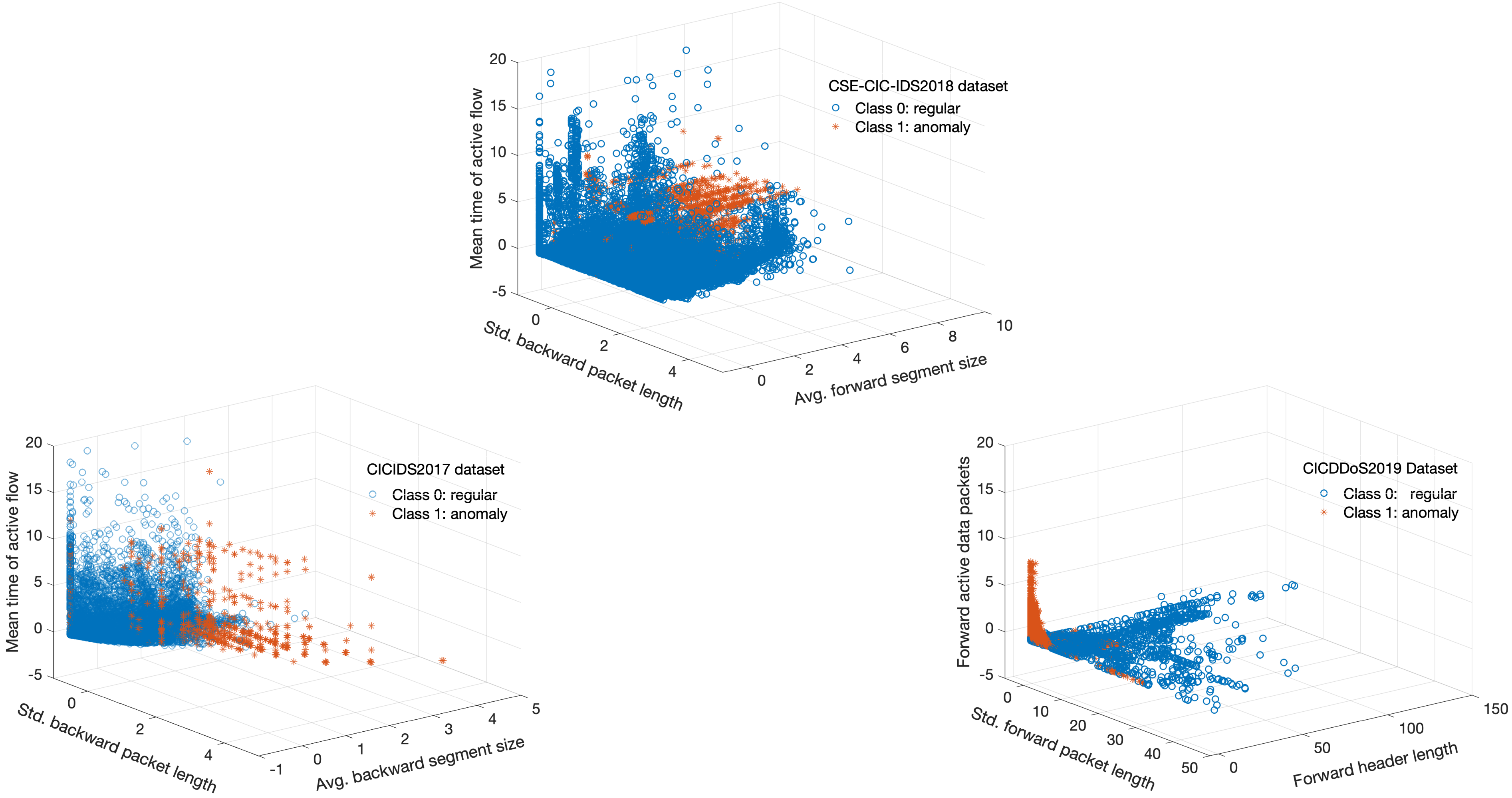
Experimental procedure

- Step 1: Use subsets of the CIC datasets to create training and test datasets
- Step 2: Normalize training and test datasets
- Step 3: Train and tune parameters of the BLS and GBDT models using time series split for 10-fold cross-validation
- Step 4: Evaluate model performance based on:
  - Training time
  - Accuracy
  - F-score
  - Precision
  - Sensitivity
  - Confusion matrix

*BLS: broad learning system

*GBDT: gradient boosting decision trees

CIC datasets: 2017, 2018, 2019



Best hyper-parameters:
BLS and incremental BLS

Model	Dataset	Mapped features	Groups of mapped features	Enhancement nodes
BLS				
RBF-BLS	CICIDS2017	20	30	40
CFBLS	CSE-CIC-IDS2018	20	10	80
BLS	CICDDoS2019	15	5	20
Incremental BLS				
CFBLS	CICIDS2017	10	20	40
BLS	CSE-CIC-IDS2018	15	30	20
CFBLS	CICDDoS2019	20	5	10

- Incremental BLS (additional parameters):
  - Incremental learning steps: 2
  - Enhancement nodes/step: 20 (CICIDS2017, CSE-CIC-IDS2018), and 10 (CICDDoS2019)
  - Data points/step: 55,680 (CICIDS2017), 49,320 (CSE-CIC-IDS2018), and 382,929 (CICDDoS2019)

Best hyper-parameters: XGBoost, LightGBM, and CatBoost

Model	Dataset	Number of estimators	Learning rate
XGBoost	CICIDS2017	100	0.01
	CSE-CIC-IDS2018	100	0.01
	CICDDoS2019	20	0.01
LightGBM	CICIDS2017	200	0.10
	CSE-CIC-IDS2018	150	0.02
	CICDDoS2019	20	0.05
CatBoost	CICIDS2017	150	0.10
	CSE-CIC-IDS2018	150	0.01
	CICDDoS2019	20	0.01

- GBDT (additional parameters):
  - Maximum depth in a tree: 6 (XGBoost, CatBoost)
  - Maximum number of leaves: 31 (LightGBM, CatBoost)
  - Loss function: log-loss
  - Boosting modes: gbtree (XGBoost), gbdt (LightGBM), and plain (CatBoost)

Best performance:
BLS and incremental BLS models

Model	Dataset	Training time	Accuracy	F-Score	Precision	Sensitivity	TP	FP	TN	FN
BLS		(s)	(%)	(%)	(%)	(%)				
RBF-BLS	CICIDS2017	37.72	96.63	96.87	97357	96.18	96,832	2,416	82,511	3,841
CFBLS	CSE-CIC-IDS2018	17.04	97.46	81.46	98.26	69.56	14,597	258	240,057	6,388
BLS	CICDDoS2019	46.64	99.98	99.99	99.99	99.99	2,541,533	204	954	220
Incremental BLS										
CFBLS	CICIDS2017	17.60	95.12	95.44	96.73	94.17	94,827	3,206	81,721	5,846
BLS	CSE-CIC-IDS2018	38.09	97.47	81.35	99.51	68.80	14,437	71	240,244	6,548
CFBLS	CICDDoS2019	79.01	99.97	99.99	99.97	99.99	2,541,764	646	512	9

Best performance:
XGBoost, LightGBM, and CatBoost models

Model	Dataset	Training time (s)	Accuracy (%)	F-Score (%)	Precision (%)	Sensitivity (%)	TP	FP	TN	FN
XGBoost	CICIDS2017	24.49	98.62	98.72	99.43	98.02	98,684	568	84,359	1,989
	CSE-CIC-IDS2018	14.43	99.90	99.39	99.99	98.79	20,731	1	240,314	254
	CICDDoS2019	62.99	99.99	99.99	99.99	99.99	2,541,767	7	1,151	6
LightGBM	CICIDS2017	3.35	97.93	98.06	99.94	96.25	96,896	60	84,867	3,777
	CSE-CIC-IDS2018	1.73	98.73	91.44	99.99	84.23	17,675	1	240,314	3,310
	CICDDoS2019	8.12	99.99	99.99	99.99	99.99	2,541,767	8	1,150	6
CatBoost	CICIDS2017	20.27	98.01	98.13	99.91	96.41	97,056	83	84,844	3,617
	CSE-CIC-IDS2018	19.03	99.95	99.72	99.97	99.46	20,872	6	240,309	113
	CICDDoS2019	17.38	99.99	99.99	99.99	99.99	2,541,762	19	1,139	11

Algorithm performance: effect of hyper-parameters

- LightGBM models offer the shortest training time for all considered datasets
- Their training time is approximately 20 times shorter than for BLS, XGBoost, and CatBoost models
- The GBDT models outperform original and incremental BLS models using the CICIDS2017 and CSE-CIC-IDS2018 datasets
- The best accuracy and F-Score:
  - XGBoost model and CICIDS2017 dataset
  - CatBoost model and CSE-CIC-IDS2018 dataset
- The lowest number of FNs is generated using XGBoost model with CICIDS2017 and CatBoost model with CSE-CIC-IDS2018 datasets
- The BLS and GBDT models using the CICDDoS2019 dataset have similar and very high accuracy, F-Score, precision, and sensitivity

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

Conclusion

- We compared performance of BLS and GBDT algorithms using CIC datasets
- Training time depends on:
  - BLS: number of mapped features, groups of mapped features, and enhancement nodes
  - GBDT: number of estimators, learning rate, maximum depth, and number of leaves in the decision trees
- The shortest training time was required for LightGBM models
- The experiments illustrated advantages of GBDT algorithms when detecting DoS and DDoS attacks

Roadmap

- Introduction
- Machine learning algorithms:
  - Broad learning system:
    BLS and its extensions with and without incremental learning
  - Gradient boosting decision trees:
    XGBoost, LightGBM, and CatBoost
- Description of datasets:
  CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019
- Experiments and performance evaluation
- Conclusion and references

References: algorithms, data sources, and tools

- Broadlearning: http://www.broadlearning.ai/
- XGBoost: https://xgboost.readthedocs.io/en/latest/
- LightGBM: https://lightgbm.readthedocs.io/en/latest/
- CatBoost: https://catboost.ai/

- CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019 datasets: https://www.unb.ca/cic/datasets/index.html

- Cedar: https://docs.computecanada.ca/wiki/Cedar
- Python: https://pypi.org
- Pandas: https://pandas.pydata.org/

References: intrusion detection

- A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, “Distributed denial of service attacks in cloud: state-of-the-art of scientific and commercial solutions,” *Computer Science Review*, vol. 39, no. 100332, Feb. 2021.
- 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 Tu*t., vol. 21, no. 1, pp. 686–728, First quarter 2019.
- G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, “Denial-of-service attack-detection techniques,” *IEEE Internet. Comput*., vol. 10, no. 1, pp. 82–89, Jan.–Feb. 2006

References: machine learning

- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning. Cambridge*, MA, USA: The MIT Press, 2016.
- K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: The MIT Press, 2012.
- C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag, 2006.

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.
- J. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–01232, Apr. 2001.
- 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 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:

- 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, to be published.
- 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.
- 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.
- 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.
- 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.