Detection of Denial of Service Attacks in Communication Networks

Ana Laura Gonzalez Rios, Zhida Li, Kamila Bekshentayeva, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
Email: {anag, zhidal, kdagilov, ljilja}@sfu.ca

***Abstract*—Detection of evolving cyber attacks is a challenging task for conventional network intrusion detection techniques. Various supervised machine learning algorithms have been implemented in network intrusion detection systems. However, traditional algorithms require long training time and have high computational complexity. Therefore, we propose detection of denial of service cyber attacks in communication networks by employing the broad learning system (BLS) that requires shorter training time while achieving comparable performance. Because designing effective detection systems relies on training and test datasets that contain anomalous network traffic data, in this paper we evaluate the performance of various BLS models by using recently generated network intrusion datasets. The best accuracy and F-Score were often achieved using BLS with cascades while BLS with incremental learning usually required shorter training time.**

***Index Terms*—Intrusion detection, network anomalies, denial of service attacks, machine learning, broad learning system**

I. Introduction

Frequent cases of cybersecurity threats that are difficult to identify and prevent are greatly affecting the Internet reliability. Primary infrastructure against these threats is based on intrusion detection systems (IDSs) [11]. Host-based systems protect the host (endpoint) by monitoring the operating system files and processes while network-based systems (NIDSs) monitor network traffic by analyzing flows of packets and inspecting packet headers [30]. Detecting malicious network intrusions may be signature-based or anomaly-based. Their role is to enhance security by identifying suspicious events in the observed network traffic. Signature-based techniques [33] rely on known events that follow similar rules and patterns while anomaly-based intrusion detection techniques [16], [39] rely on detecting deviations from an expected behavior.

Various NIDSs [8], [25] have been recently proposed to address a dynamically changing landscape of cyber threats. They employ diverse deep learning algorithms [10], [18], [35] such as convolutional neural networks, recurrent neural networks (RNNs) [14], [20], deep belief networks, and autoencoders that offer promising results for anomaly detection [23], [24], [36]. Training time for deep neural networks may be reduced by selecting appropriate features and parameters while still maintaining high accuracy [45]. Combination of supervised learning and feature selection algorithms is employed to devise novel intrusion detection solutions that address the high false alarm rate by classifying previously unobserved network traffic patterns [43]. Reported results demonstrate that the proposed anomaly-based IDS employing a neural network with a wrapper feature selection outperforms other models. A deep neural network model with several hidden layers yielding high accuracy has been also introduced [21]. The proposed systems have been evaluated using KDD Cup 99 [4], [32], [42] and NSL-KDD [3], [44] datasets.

Support Vector Machine (SVM), a widely used machine learning algorithm, was employed to identify Border Gateway Protocol (BGP) anomalies [15], [27]. Various types of supervised RNNs (long short-term memory (LSTM) [20] and gated recurrent unit (GRU) [14]) as well as the broad learning system (BLS) [12], [13], [31] have been also evaluated to compare their performance when detecting anomalies in network traffic [9], [17], [26], [28], [29], [38]. Performance of the proposed methods often depends on selected features and their combinations. Hence, algorithms such as decision tree were used to extract relevant features for detecting anomalies using SVM and LSTM. The main disadvantages of conventional machine learning techniques are long training time and computational complexity. In contrast, BLS employs fewer hidden layers, relies on calculating pseudo-inverse during the training process, and requires comparably shorter training time when used for function approximation, time series forecast, and image recognition [12], [13], [22].

Performance of an NIDS depends on reliable training and test datasets. The scarcity of adequate data is a research challenge because the behavioral patterns of intrusion instances evolve. A cross-validation comparative study [34] indicated degraded performance of algorithms trained using KDD Cup 99 when tested with newer datasets. Therefore, we consider the effectiveness of the employed algorithms based on datasets from the Canadian Institute for Cybersecurity (CIC) Intrusion Detection System (CICIDS2017) [1] and the collaborative project between the Communications Security Establishment (CSE) and the CIC (CSE-CIC-IDS2018) [2] that contain the latest denial of service (DoS) attacks. We evaluate performance of BLS and its extensions as alternative supervised learning algorithms for detecting network anomalies [28], [29].

The paper is organized as follows: After introducing the topic and related work, descriptions of experimental testbeds

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant R31-611284.

and datasets are provided in Section II. The BLS algorithm and its extensions are presented in Section III. The experimental procedure and performance evaluation are given in Section IV. We conclude with Section V.

II. Intrusion Detection Testbeds and Datasets

Performance of an NIDS may be evaluated using simulation tools, emulators, or testbeds. Testbeds consist of firewalls, routers, switches, and operating systems. CIC has developed a testbed framework [40], [41] to generate CICIDS2017 [1] and CSE-CIC-IDS2018 [2] traffic data. The CICIDS2017 testbed includes an attacker-network consisting of one router, one switch, and four terminals with Kali Linux and Windows 8.1 operating systems. The victim-network consists of three servers, one firewall, two switches, and ten terminals interconnected by a security authentication server. One switch in the victim-network serves as a mirror port and captures the incoming and outgoing traffic. The CSE-CIC-IDS2018 [2] attacker-network includes 50 terminals while the victim-network is implemented as a Local Area Network (LAN) with 420 terminals and 30 servers distributed over five subnets. Ubuntu, Windows 8.1, and Windows 10 operating systems are installed on host machines while servers use Windows 2012 and Windows 2016. Both networks were implemented using the Amazon Web Services.

Benign (regular) and attack (anomalous) data are systematically generated using profiles. The benign data are obtained using the B-Profiles that generate background traffic based on the analysis of user behavior when executing application protocols in a non-malicious manner. The simulated protocols include: HTTP, HTTPS, SMTP, POP3, IMAP, SSH, and FTP. After creating the B-Profile, a Java agent generates benign events. The M-Profile captures details of the most common types of attacks such as brute force, botnet, DoS, DDoS, heartbleed, infiltration, and web attack. These attack scenarios may be interpreted either by network operators or autonomous systems.

The CICIDS2017 dataset includes intrusions that rely on various network vulnerabilities [40] and are executed using attack tools: Patator, Slowloris, Heartleech, Damn Vulnerable Web App, Metasploit, Ares, and Low Orbit Ion Cannon. Extraction of 84 features including duration, size of packets, number of packets, and number of bytes was performed using CICFlowMeter, an application for generating and analyzing network traffic flows [5]. We use DoS data collected on Wednesday, July 05, 2017. Data points are labeled GoldenEye, Hulk, SlowHTTPTest, and Slowloris having 10,293, 230,124, 5,499, and 5,796 intrusions, respectively.

The CSE-CIC-IDS2018 dataset was captured over ten business days between Wednesday, February 14, 2018 and Friday, March 02, 2018 [2]. It includes type, date, and start and end times of the attacks. Extracted are 83 features including flow duration, minimum/maximum packet size, and destination port. We consider GoldenEye and Slowloris DoS attacks generated on Thursday, February 15, 2018 from 09:26 to 10:09 and from 10:59 to 11:40, respectively.

III. Broad Learning System and its Extensions

BLS [6] offers shorter training time and comparable performance by using a single layer feedforward neural network and pseudo-inverse to calculate outputs. Several BLS extensions exploit flexible structure of the algorithm and include: incremental learning [12], radial basis function (RBF) network (RBF-BLS) [31] as well as cascades of mapped features (CFBLS), enhancement nodes (CEBLS), and both mapped features and enhancement nodes (CFEBLS) [13].

BLS improves the random vector functional-link neural network [37] by mapping the input data $\boldsymbol{X}$ to a set of mapped features $\boldsymbol{Z}^n \triangleq [\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n]$ that generates enhancement nodes $\boldsymbol{H}^m \triangleq [\boldsymbol{H}_1, ..., \boldsymbol{H}_m]$ using random weights. Groups of mapped features and enhancement nodes are defined as:

$$\boldsymbol{Z}_i = \phi(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i}), i = 1, 2, ..., n \tag{1}$$

$$\boldsymbol{H}_j = \xi(\boldsymbol{Z}_x^n \boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}), j = 1, 2, ..., m, \tag{2}$$

where ϕ $(linear)$ and ξ $(tanh)$ are the feature and enhancement mappings, respectively. $\boldsymbol{W}_{e_i}$ and $\boldsymbol{W}_{h_j}$ are weights, and $\boldsymbol{\beta}_{e_i}$ and $\boldsymbol{\beta}_{h_j}$ are bias parameters. A state matrix $\boldsymbol{A}_n^m$ is constructed by concatenating matrices $\boldsymbol{Z}^n$ and $\boldsymbol{H}^m$ associated with n groups of mapped features and m groups of enhancement nodes, respectively. The Moore-Penrose pseudo-inverse of matrix $\boldsymbol{A}_n^m$ is computed to calculate the weights $\boldsymbol{W}_n^m$ for the given output $\boldsymbol{Y}$. During testing, data labels are predicted using the calculated weights, mapped features, and enhancement nodes.

The BLS structure may be dynamically expanded by using incremental learning to include the additional input data $\boldsymbol{X}_a$, mapped features $\boldsymbol{Z}_{n+1}$, and enhancement nodes $\boldsymbol{H}_{m+1}$. It requires shorter training time because the weights are updated using only the incremental input data instead of retraining the entire model. RBF-BLS employs Gaussian function as the enhancement mapping ξ. The structure of BLS with cascades is defined by the connections within and between the mapped features and enhancement nodes. The CFBLS and CEBLS architectures are shown in Fig. 1. In the case of CFBLS, the new group (k) of mapped features is created by using the previous group $(k-1)$. The groups of mapped features are formulated as:

$$\begin{aligned}\boldsymbol{Z}_k &= \phi(\boldsymbol{Z}_{k-1}\boldsymbol{W}_{e_k} + \boldsymbol{\beta}_{e_k}) \\ &\triangleq \phi^k(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^k), \text{for } k = 1, ..., n.\end{aligned} \tag{3}$$

The cascades of these groups $\boldsymbol{Z}^n \triangleq [\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n]$ are used to generate the enhancement nodes $\{\boldsymbol{H}_j\}_{j=1}^m$. The first CEBLS enhancement node is generated from mapped features while subsequent nodes are generated from previous nodes creating a cascade:

$$\begin{aligned}\boldsymbol{H}_u &= \xi(\boldsymbol{H}_{u-1}\boldsymbol{W}_{e_u} + \boldsymbol{\beta}_{e_u}) \\ &\triangleq \xi^u(\boldsymbol{Z}^n; \{\boldsymbol{W}_{h_i}, \boldsymbol{\beta}_{h_i}\}_{i=1}^u), \text{for } u = 1, ..., m,\end{aligned} \tag{4}$$

where $\boldsymbol{W}_{h_i}$ and $\boldsymbol{\beta}_{h_i}$ are randomly generated. The CFEBLS architecture is a combination of the two cascading approaches. The structure of incremental CFEBLS is shown in Fig. 2.

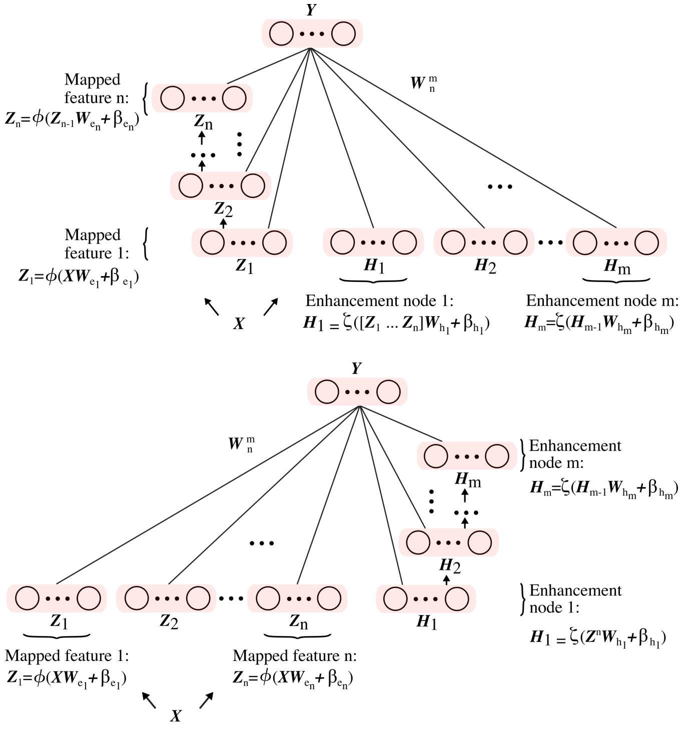
Y
W^{m}_{n}
Mapped feature n: $Z_n=\phi(Z_{n-1}W_{e_n}+\beta_{e_n})$
Z_n
Z_2
Mapped feature 1: $Z_1=\phi(XW_{e_1}+\beta_{e_1})$
Z_1
H_1
H_2
H_m
X
Enhancement node 1: $H_1=\zeta([Z_1 \ldots Z_n]W_{h_1}+\beta_{h_1})$
Enhancement node m: $H_m=\zeta(H_{m-1}W_{h_m}+\beta_{h_m})$
Y
W^{m}_{n}
Enhancement node m: $H_m=\zeta(H_{m-1}W_{h_m}+\beta_{h_m})$
H_m
H_2
Z_1
Z_2
Z_n
H_1
Enhancement node 1: $H_1=\zeta(Z^nW_{h_1}+\beta_{h_1})$
Mapped feature 1: $Z_1=\phi(XW_{e_1}+\beta_{e_1})$
Mapped feature n: $Z_n=\phi(XW_{e_n}+\beta_{e_n})$
X

Fig. 1. Modules of the CFBLS (top) and CEBLS (bottom) algorithms. Shown are cascades of mapped features (top) and enhancement nodes (bottom) without incremental learning.

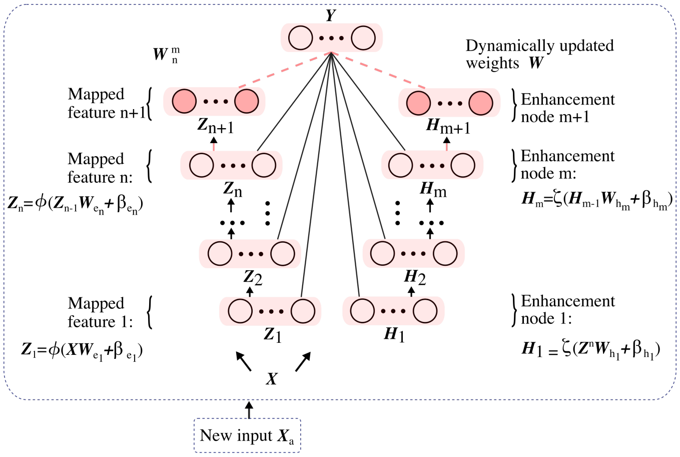

Fig. 2. Module of the CFEBLS algorithm with increments of mapped features $\boldsymbol{Z}_{n+1}$, enhancement nodes $\boldsymbol{H}_{m+1}$, and new input data $\boldsymbol{X}_{a}$.

IV. Experimental Procedure and Performance Evaluation

We implement and evaluate performance of RBF-BLS and BLS with cascades of mapped features (CFBLS), enhancement nodes (CEBLS), and both mapped features and enhancement nodes (CFEBLS) with and without incremental learning. We perform two-way classification to identify regular (0) and anomalous (1) data using subsets of CICIDS2017 and CSE-CIC-IDS2018 datasets that include application-layer DoS attacks. Traffic data collected on Wednesday, July 05, 2017 and Thursday, February 15, 2018 are used to create the training and test datasets consisting of 60 % and 40 % of anomalous data, respectively. Before partitioning, the data points are sorted based on time stamps. The experimental procedure consists of the four steps shown in Fig. 3: (1) Extracting the CICIDS2017 and CSE-CIC-IDS2018 subsets for training and testing; (2) Removing invalid data ("NaN" and "infinity"), converting categorical to numerical features using dummy coding, and normalizing training and test datasets to have mean 0 and standard deviation 1 employing the *zscore* function; (3) Using 10-fold validation to train and tune parameters; and (4) Testing and evaluating generated machine learning (ML) models based on accuracy, F-Score, and training time.

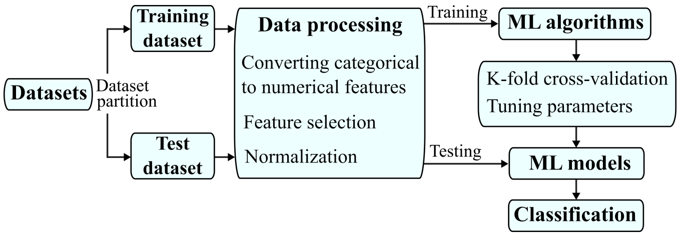

Fig. 3. Experimental procedure for network intrusion detection.

We experiment with 78 features and subsets of top 2^n (n = 3, 4, 5, and 6) features to evaluate their effect on BLS performance. Features are ranked using the extremely randomized (extra) trees [19] that introduce splitting the nodes based on a random value in order to avoid over-fitting. The features are ranked based on importance using *sklearn.ensemble.ExtraTreesClassifier()* [7] function. We use parameters $n\_estimators = 100$, $random\_state = 1$, and default values for the remaining settings. The sixteen most relevant features and their importance are shown in Table I.

TABLE I
Sixteen Most Relevant Features and Their Importance

CICIDS2017	
1. Dst Port (0.0595)	2. Pkt Size Avg (0.0464)
3. Bwd Pkt Len Mean (0.0462)	4. Flow IAT Max (0.0440)
5. Protocol (0.0437)	6. Pkt Len Std (0.0423)
7. Bwd Seg Size Avg (0.0418)	8. ACK Flag Cnt (0.0416)
9. Fwd IAT Max (0.0403)	10. Fwd IAT Std (0.0345)
11. Fwd Seg Size Min (0.0329)	12. Idle Max (0.0319)
13. Fwd Pkts/s (0.0296)	14. Bwd Pkt Len Max (0.0287)
15. Idle Mean (0.0264)	16. Bwd Pkt Len Std (0.0256)
CSE-CIC-IDS2018	
1. Fwd Seg Size Min (0.2904)	2. Init Fwd Win Byts (0.1082)
3. Bwd Pkt Len Std (0.0474)	4. Bwd Pkt Len Max (0.0306)
5. Pkt Len Max (0.0264)	6. Flow IAT Min (0.0254)
7. Flow Duration (0.0248)	8. Fwd IAT Tot (0.0218)
9. Fwd IAT Min (0.0206)	10. Bwd Pkt Len Mean (0.0197)
11. ACK Flag Cnt (0.0190)	12. Init Bwd Win Byts (0.0190)
13. Fwd IAT Max (0.0185)	14. PSH Flag Cnt (0.0167)
15. Dst Port (0.0164)	16. Flow IAT Max (0.0162)

Performance of BLS models based on the number of selected features is shown in Fig. 4. As expected, shorter training time is required for models using fewer number of features and models based on the incremental BLS. Selecting only 8 features generated accuracy and F-Score over 90 % for CICIDS2017 dataset. No such advantage is observed for the F-Score using the CSE-CIC-IDS2018 dataset. Using 32 features shows comparable results for both datasets while taking shorter training time compared to using 64 or 78 features. Non-incremental and incremental model parameters leading to the best performance are shown in Table II while accuracy and F-Score are shown in Table III. Most generated models achieve accuracy and F-Score above 96 % and 90 %, respectively. Our past results when applying non-incremental and incremental BLS to BGP anomaly data were comparable [29]. Experiments are performed using a Dell Alienware Aurora with 32 GB memory and Intel Core i7 7700K processor. Results are generated using Python 3.6.

TABLE II
Best Parameters Obtained when Using CICIDS2017 and CSE-CIC-IDS2018 Datasets; Incremental Learning: *incremental learning steps* = 2, *enhancement nodes/step* = 20, and *data points/step* = 55,680 (CICIDS2017), 49,320 (CSE-CIC-IDS2018)

Parameters	**Number of features**		
	78	**64**	**32**
Non-incremental BLS		**CICIDS2017**	
Model	RBF-BLS	BLS	CEBLS
Mapped features	20	10	10
Groups of mapped features	30	30	10
Enhancement nodes	40	20	40
Incremental BLS			
Model	CFBLS	CFEBLS	CFBLS
Mapped features	10	20	10
Groups of mapped features	20	20	20
Enhancement nodes	40	20	40
Non-incremental BLS		**CSE-CIC-IDS2018**	
Model	CFBLS	RBF-BLS	CEBLS
Mapped features	20	20	15
Groups of mapped features	10	10	20
Enhancement nodes	80	80	80
Incremental BLS			
Model	BLS	CEBLS	BLS
Mapped features	15	20	10
Groups of mapped features	30	10	20
Enhancement nodes	20	40	20

TABLE III
Best Performance for Models Tested with CICIDS2017 and CSE-CIC-IDS2018 Datasets

Features	**Dataset**	**Accuracy**	**F-Score**	**Model**	**Training time**
Non-incremental BLS		**(%)**	**(%)**		**(s)**
78	CICIDS2017	96.63	96.87	RBF-BLS	15.60
	CSE-CIC-IDS2018	97.46	81.46	CFBLS	4.13
64	CICIDS2017	96.10	96.35	BLS	8.97
	CSE-CIC-IDS2018	98.60	90.49	RBF-BLS	4.65
32	CICIDS2017	96.34	96.62	CEBLS	39.25
	CSE-CIC-IDS2018	98.83	92.26	CEBLS	33.46
Incremental BLS					
78	CICIDS2017	95.12	95.44	CFBLS	3.69
	CSE-CIC-IDS2018	97.47	81.35	BLS	6.78
64	CICIDS2017	94.88	95.38	CFEBLS	7.39
	CSE-CIC-IDS2018	96.70	74.64	CEBLS	11.59
32	CICIDS2017	95.39	95.75	CFBLS	6.39
	CSE-CIC-IDS2018	97.08	77.89	BLS	5.65

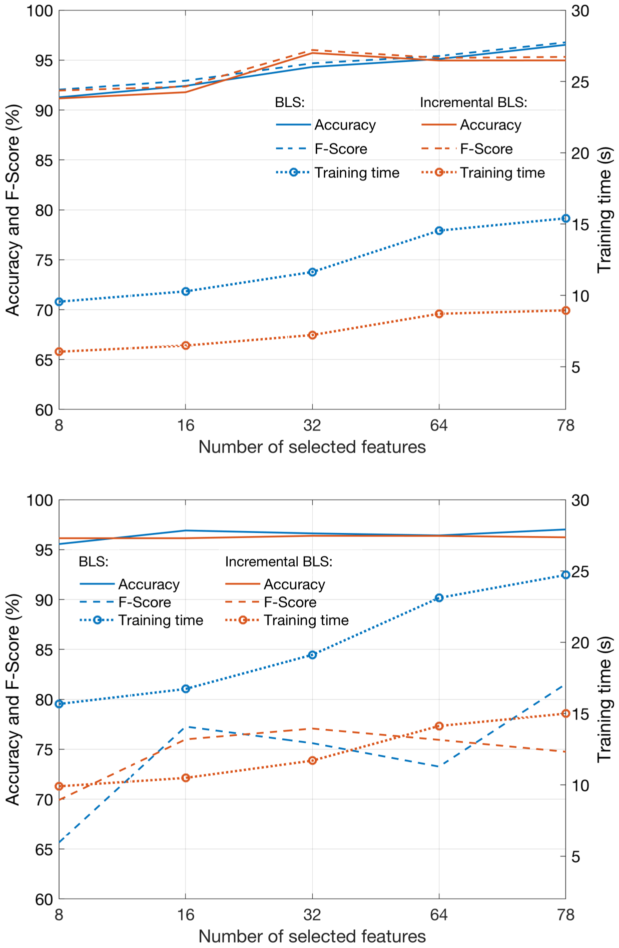

Fig. 4. Performance of BLS (blue) and incremental BLS (red). Shown are accuracy, F-Score, and training time for 78 features and subsets of 2^n ($n = 3, 4, 5, and\ 6$) most relevant features using CICIDS2017 (top) and CSE-CIC-IDS2018 (bottom) datasets.

V. Conclusions

We considered malicious intrusions and anomalies in communication networks and evaluated performance of machine learning algorithm such as BLS, RBF-BLS as well as BLS with cascades of mapped features and cascades of enhancement nodes with and without incremental learning. Experiments were conducted using subsets of CICIDS2017 and CSE-CIC-IDS2018 datasets that contained DoS attacks captured from experimental testbeds. The developed models generated comparable performance even when selecting a smaller number of relevant features. The BLS learning models were compared based on accuracy, F-Score, and training time. Larger number of mapped features, groups of mapped features, and enhancement nodes required additional memory and longer training time. Experimental results indicated that most generated models achieved accuracy and F-Score above 90 %. While CEBLS and CFEBLS models required longer training time, the training time for incremental BLS was significantly shorter than for non-incremental BLS because models did not need to be retrained.

References

[1] Intrusion Detection Evaluation Dataset (CICIDS2017) [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. Accessed: Feb. 7, 2020.

[2] A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018) [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018. Accessed: Feb. 7, 2020.

[3] NSL-KDD Dataset [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html. Accessed: Feb. 7, 2020.

[4] KDD Cup 1999 Data [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed: Feb. 7, 2020.

[5] CICFlowMeter [Online]. Available: http://netflowmeter.ca/netflowmeter.html. Accessed: Feb. 7, 2020.

[6] Broadlearning [Online]. Available: http://www.broadlearning.ai. Accessed: Feb. 7, 2020.

[7] Sklearn.ensemble.ExtraTreesClassifier [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTrees Classifier.html. Accessed: Feb. 7, 2020.

[8] W. Alhakami, A. Alharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, "Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52181–52190, Apr. 2019.

[9] S. A. Althubiti, E. M. Jones Jr., and K. Roy, "LSTM for anomaly-based network intrusion detection," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf.*, Sydney, Australia, Nov. 2018, pp. 1–3.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag, 2006.

[11] 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.

[12] 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.

[13] 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.*, pp. 1–14, Sept. 2018.

[14] 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.

[15] 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*, A. Dehghantanha, M. Conti, and T. Dargahi, Eds., Berlin: Springer, 2018, pp. 47–70.

[16] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Comput. Secur.*, vol. 28, pp. 18–28, Feb.-Mar. 2009.

[17] A. L. Gonzalez Rios, Z. Li, G. Xu, A. Diaz Alonso, and Lj. Trajković, "Detecting network anomalies and intrusions in communication networks," in *Proc. 23rd IEEE Int. Conf. Intell. Eng. Syst.*, Gödöllö, Hungary, Apr. 2019, pp. 29–34.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: The MIT Press, 2016.

[19] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.

[20] 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.

[21] Y. Jia, M. Wang, and Y. Wang, "Network intrusion detection algorithm based on deep neural network," *IET Inform. Secur.*, vol. 13, no. 1, pp. 48–53, Jan. 2019.

[22] J.-W. Jin and C. L. P. Chen, "Regularized robust broad learning system for uncertain data modeling," *Neurocomputing*, vol. 322, pp. 58–69, Dec. 2018.

[23] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," in *Proc. Int. Congr. Big Data, Deep Learning Fighting Cyber Terrorism*, Ankara, Turkey, Dec. 2018, pp. 113–116.

[24] T. Kim, S. C. Suh, H. Kim, J. Kim, and J. Kim, "An encoding technique for CNN-based network anomaly detection," in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, Dec. 2018, pp. 2960–2965.

[25] M. Labonne, A. Olivereau, B. Polvé, and D. Zeghlache, "A cascade-structured meta-specialists approach for neural network-based intrusion detection," in *Proc. 16th IEEE Annu. Consumer Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2019, pp. 1–6.

[26] D. Lavrova, D. Zegzhda, and A. Yarmak, "Using GRU neural network for cyber-attack detection in automated process control systems," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw.*, Sochi, Russia, June 2019, pp. 1–3.

[27] 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.

[28] 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.

[29] 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 Syst.*, Sapporo, Japan, May 2019.

[30] M. C. Libicki, L. Ablon, and T. Webb, *The Defender's Dilemma: Charting a Course Toward Cybersecurity*. Santa Monica, CA, USA: RAND Corporation, 2015.

[31] Z. Liu and C. L. P. Chen, "Broad learning system: structural extensions on single-layer and multi-layer neural networks," in *Proc. Int. Conf. Secur., Pattern Anal., Cybern.*, Shenzhen, China, Dec. 2017, pp. 136–141.

[32] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inform. Syst. Security*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[33] 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.

[34] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inform. Secur. J.: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Jan. 2016.

[35] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: The MIT Press, 2012.

[36] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, Aug. 2018.

[37] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, Apr. 1994.

[38] Z. Qu, L. Su, X. Wang, S. Zheng, X. Song, and X. Song, "A unsupervised learning method of anomaly detection using GRU," *2018 IEEE Int. Conf. Big Data Smart Comput.*, Shanghai, China, Jan. 2018, pp. 685–688.

[39] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *Proc. Int. Conf. Elect., Electron., Commun., Comput., Optim. Techn.*, Mysuru, India, Dec. 2017, pp. 141–147.

[40] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *J. Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, July 2017.

[41] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inform. Syst. Secur. Privacy*, Funchal, Portugal, Jan. 2018, pp. 108–116.

[42] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: results from the JAM project," in *Proc. DARPA Inform. Survivability Conf. Expo.*, Hilton Head, SC, USA, Jan. 2000, pp. 130–144.

[43] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *Proc. Int. Conf. Robotics, Elect. Signal Process. Techn.*, Dhaka, Bangladesh, Jan. 2019, pp. 643–646.

[44] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, July 2009, pp. 1–6.

[45] J. Woo, J. Song, and Y. Choi, "Performance enhancement of deep neural network using feature selection and preprocessing for intrusion detection," in *Proc. Int. Conf. Artif. Intell. Inform. Commun.*, Okinawa, Japan, Feb. 2019, pp. 415–417.