Detection of Denial of Service Attacks Using Echo State Networks

Kamila Bekshentayeva and Ljiljana Trajković

Abstract— **Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are major threats to cybersecurity in communication networks. These cyber attacks are evolving and becoming more difficult to identify and, hence, a number of intrusion detection approaches have been proposed. Various machine learning techniques have proved useful in detecting such anomalies. We rely on supervised machine learning and apply echo state networks to detect known DoS and DDoS attacks. Echo state networks belong to a reservoir computing approach used to train recurrent neural networks. Their performance is compared to bidirectional long short-term memory using datasets collected by the Canadian Institute for Cybersecurity and the RIPE and Route Views data collection sites. Performance is evaluated based on accuracy, F-Score, false alarm rate, and training time. Experimental results indicate that echo state networks have comparable performance and shorter training time.**

I. Introduction

Denial of Service (DoS) attacks are attempts of an attacker to make services unavailable to legitimate users. Distributed DoS (DDoS) attacks combine multiple compromised end systems in a coordinated way to exhaust resources of a targeted system [14]. Two general approaches for detecting such attacks are classified as signature-based and anomaly-based. Recent events indicate that the DoS and DDoS attacks are becoming more sophisticated and, hence, more difficult to detect especially when using signature-based countermeasures. The continuous growth of vulnerable and interconnected end systems increases occurrences of successful DDoS attacks [14]. Hence, defence mechanisms against DoS and DDoS attacks have received considerable attention in the area of cybersecurity.

Various machine learning algorithms have been employed for detecting network anomalies [12], [18], [19], [24]. Broad learning system (BLS) [13] models were evaluated for detection of DoS attacks [16]. Successful detection of network traffic anomalies using CIC-IDS and Border Gateway Protocol (BGP) datasets [20], [21] was also achieved with boosting algorithms such as light gradient boosting machine (LightGBM) and recurrent neural networks (RNNs) including long short-term memory (LSTM) and gated recurrent unit (GRU).

Reservoir computing (RC) is an approach for supervised training of RNNs. It does not experience vanishing and exploding gradients because training is performed to obtain only optimal output weights. We apply Echo State Networks (ESNs) as a feasible RC approach to identify network intrusions by employing binary classification of regular and anomalous data points. ESNs have been employed in a variety of domains and tasks [17], [22] including time series forecasting, wireless communication networks, speech and handwriting recognition, music imitation, robot control, and network anomaly detection. Online anomaly detection systems deal with streams of input data in real time. Due to their computational efficiency, ESNs have been used in an online anomaly detection framework implemented using sensor networks [12].

Machine learning-based models utilize datasets that reflect the characteristics of diverse network behaviour. Synthetically generated CIC-IDS network connection records from the Canadian Institute for Cybersecurity (CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019) [2] are employed to evaluate ESN models. We also consider the BGP datasets collected by the RIPE [5] and Route Views [7] data collection sites. They capture BGP worm attacks that occurred in 2001 and 2003 as well as large DDoS attacks in 2019 [3], [8] and 2020 [1].

Computationally efficient anomaly detection techniques have been employed to deal with large input datasets [11]. ESNs are a computationally efficient approach to detect network anomalies and achieve comparable performance. We evaluate the influence of hyperparameters on performance of ESN models. A variation of k-fold cross-validation for time series [9] is used to evaluate the selection of hyperparameters. The best performing ESN model is used for comparison with bidirectional LSTM (Bi-LSTM), a widely used recurrent neural networks.

The paper is organized as follows: After introducing the topic in Section I, we describe ESNs in Section II and introduce datasets in Section III. The experimental procedure is given in Section IV while performance evaluation is discussed in Section V. We conclude with Section VI.

II. Echo State Networks

The reservoir in ESNs is a randomly connected network with input $\mathbf{x}(n) \in \mathcal{R}^{N_x}$ and output $\mathbf{y}(n) \in \mathcal{R}^{N_y}$ that should match the given labels $\mathbf{y}^{target}(n) \in \mathcal{R}^{N_y}$ in order to minimize the loss. Reservoir parameters are randomly generated input weight matrix $\mathbf{W}^{in} \in \mathcal{R}^{N_x \times N_z}$ and randomly generated reservoir weight matrix $\mathbf{W} \in \mathcal{R}^{N_z \times N_z}$ [23], where N_x and N_y are numbers of input and output nodes, respectively. Reservoir state equations are:

$$\widetilde{\mathbf{z}}(n) = tanh(\mathbf{W}^{in} \times \mathbf{x}(n) + \mathbf{W} \times \mathbf{z}(n-1)) \tag{1}$$

$$\mathbf{z}(n) = (1-\alpha)\mathbf{z}(n-1) + \alpha\widetilde{\mathbf{z}}(n), \tag{2}$$

where $\widetilde{\mathbf{z}}(n) \in \mathcal{R}^{N_z}$ is reservoir's update at discrete time $n = \{1, ..., N\}$, N is the number of data points in the training

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

The authors are with the School of Engineering Science, Simon Fraser University, Vancouver, British Columbia, Canada V5A 1S6, {kdagilov, ljilja}@sfu.ca

dataset, N_z is the number of reservoir nodes, $tanh(\cdot)$ is hyperbolic tangent, $\mathbf{z}(n) \in \mathcal{R}^{N_z}$ is the reservoir state, and $\alpha \in (0,1]$ is the leaking rate. When $\alpha = 1$, $\mathbf{z}(n) \equiv \widetilde{\mathbf{z}}(n)$ and the reservoir has no memory of its previous state.

ESNs are said to have the echo state property if they are able to "wash out" the initial state of the reservoir at a rate independent of the input sequence. Hence, the effect of the past input on the reservoir gradually fades away [17].

ESNs are computationally non-expensive because they do not employ gradient-based iterative optimization algorithms such as backpropagation to compute the optimal weights [17]. The difference between gradient-based and RC-based RNN approaches is shown in Fig. 1 In the case of gradient-based training, the computed optimal weights are $\mathbf{W}^{in}$, $\mathbf{W}$, and $\mathbf{W}^{out}$ are updated iteratively while in the case of RC, the output weights $\mathbf{W}^{out}$ are calculated in a single iteration.

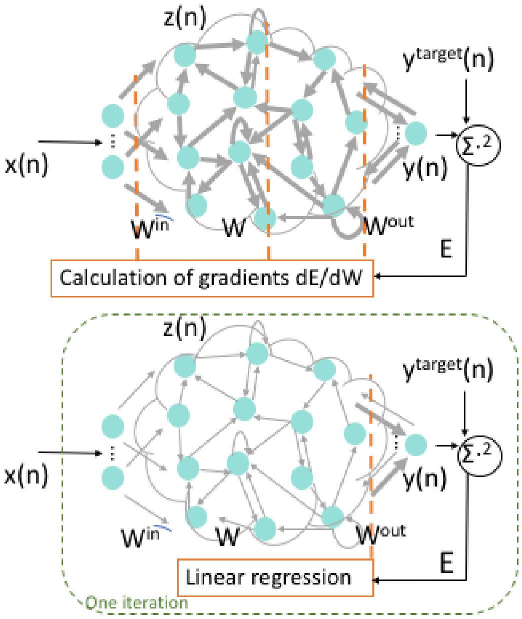

Fig. 1. Gradient-based (top) and RC-based (bottom) RNN training: dE/dW is gradient of loss function E with respect to weights. The operator $\sum(\cdot)^2$ indicates the sum of squared errors between the output and target vectors. The computation of optimal weights is indicated in bold.

A. Hyperparameters of the ESN Reservoir

Reservoir in ESNs serves as a memory of the input that becomes linearly separable using the nonlinear high-dimensional expansion $\mathbf{z}(n) \in R^{N_z}$.

The ESNs hyperparameters are: reservoir size N_z, reservoir sparsity, spectral radius $\rho(\mathbf{W})$, scaling parameter for $\mathbf{W}^{in}$, and leaking rate α [23]. *Reservoir size* or number of nodes N_z determines the ESNs memory capacity. *Reservoir sparsity* reflects the number of zero elements. The reservoir weights $\mathbf{W}$ are usually sparse because such connections yield better performance [23]. *Spectral radius* $\rho(\mathbf{W})$ is the maximal eigenvalue $|\lambda|$ of a random sparse matrix $\mathbf{W}$. A larger value of spectral radius is employed for the tasks where a longer history of the input is required. If the output $\mathbf{y}(n)$ relies on a more recent history of the input $\mathbf{x}(n)$, a smaller value of the radius is recommended. Selecting spectral radius $\rho(\mathbf{W}) \leq 1$ assures the echo state property [23] of an ESN model. *Input weights scaling* is used to scale down large input weights and enable the reservoir to be driven more by its dynamics rather than the input (1). *Leaking rate* α is related to the reservoir's update dynamics. The reservoir state values (2) change more gradually for a lower α that induces slow dynamics of $\mathbf{z}(n)$ thus increasing the duration of the short-term memory of the network.

B. Training Echo State Networks

Root mean-squared error $E(\mathbf{y}, \mathbf{y}^{target})$ is the loss function that is minimized with respect to parameters:

$$E(\mathbf{y}, \mathbf{y}^{target}) = \frac{1}{N_y} \sum_{n=1}^{N_y} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i(n) - y_i^{target}(n))^2}. \tag{3}$$

Ridge regression (4) is the most preferred option [23] to calculate optimal output weights:

$$\mathbf{W}^{out} = (\mathbf{Z}^T\mathbf{Z} + \beta\mathbf{I})^{-1}\mathbf{Z}^T\mathbf{Y}^{target}, \tag{4}$$

where matrix $\mathbf{Z} \in \mathcal{R}^{N \times (N_z + N_x)}$ is generated by horizontally concatenating the column vectors $[\mathbf{z}(n); \mathbf{x}(n)]$ for all training data points n, β is regularization coefficient used to reduce overfitting usually indicated by large output weights, and $\mathbf{I}$ is the identity matrix. Labels $\mathbf{y}^{target}(n) \in \mathcal{R}^{N_y}$ are used to form the matrix $\mathbf{Y}^{target} \in \mathcal{R}^{N \times N_y}$.

Adding scaled white noise to the input $\mathbf{x}(n)$ is also used for regularization [10]. The training time may be reduced by selecting a smaller reservoir and/or smaller dataset [23]. However, this may adversely affect model's generalization ability [10].

The ESN structure shown in Fig. 2 includes input $\mathbf{X}$, reservoir state $\mathbf{Z}$, output $\mathbf{Y}$, random input $\mathbf{W}^{in}$, random reservoir $\mathbf{W}$, optional feedback $\mathbf{W}^{fb}$, and trainable output $\mathbf{W}^{out}$ weight matrices. The symbol $\int$ represent non-linear transformation; $\mathbf{W}^{fb}$ and adjacent $(n-1)$ unit delay are optional feedback.

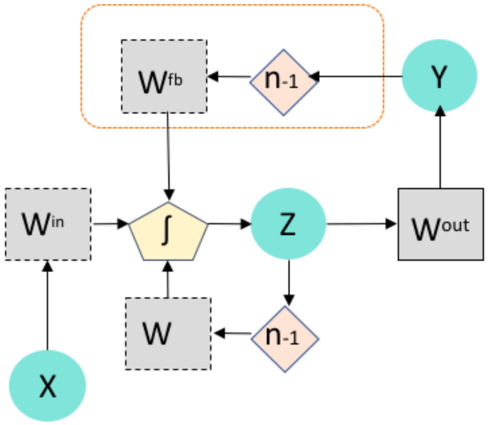

Fig. 2. High level of the ESN structure with optional feedback.

III. Network Intrusion and BGP Datasets

Performance evaluation of network intrusion detection models relies on datasets containing diverse traffic and features. We use CIC-IDS network connection records and BGP datasets that were acquired from BGP trace collectors.

A. *Network Intrusion Datasets: Attacks and Features*

The CIC synthetic testbed framework is used to generate the CIC-IDS datasets [2]: CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019 datasets. Two profile classes were used: B-profiles incorporate the abstract behavior of users based on the most frequently used protocols while M-profiles generate well-known attacks scenarios. We select GoldenEye, Hulk, SlowHTTPTest, and Slowloris attacks from CICIDS2017 collected on Wednesday, July 5, 2017; GoldenEye and Slowloris attacks from CSE-CIC-IDS2018 collected on Thursday, February 15, 2018; and LDAP, NTP, SYN, UDP-lag, and WebDDoS attacks from CICDDoS2019 dataset collected on Saturday, December 1, 2018.

Packet length features are valuable in detecting volumetric amplification DDoS attacks (floods) and monopolizing application layer DoS attacks. Standard deviation of packet length helps differentiate regular from anomalous traffic. Regular traffic exhibits high variation in packet length while the length of malicious packets is often small in the case of TCP state exhaustion attacks such as SYN and ICMP attacks. Moreover, the attackers often generate fixed-sized packets and, hence, the minimum average segment size in a malicious flow may be smaller than in regular flows.

TCP flags ACK, SYN, and URG are often used by attackers to disrupt the target's regular operation. Thus, features such as "ACK Flag Count", "SYN Flag Count", and "URG Flag Count" may help detect malicious traffic. SYN attacks may bring down a network connection through a large number of seemingly legitimate TCP requests with SYN and ACK flags set to 1, as shown in Fig. 3. Hence, the server becomes unable to respond to legitimate requests in the absence of available connections.

B. *BGP Datasets: Attacks and Features*

BGP [6] plays an essential role in routing Internet traffic between Autonomous Systems. Archives with BGP routing information are publicly available from the RIPE [5] and Route Views [7] data collection sites. Slammer, Nimda, and Code Red I worms that caused DoS attacks are downloaded from RIPE (collector rrc04 at CIXP, Geneva, Switzerland) [15]. Data capturing the Amazon Web Services (AWS) DDoS attacks (DDoS2019, DDoS2020) are downloaded from RIPE (collector rrc14, Palo Alto, CA, USA) and Route Views (collector route-views4, Eugene, OR, USA). Datasets are generated by extracting 37 AS-path and volume features. Features extracted from BGP update messages include: number of announcements, withdrawals, announced network layer reachability information (NLRI) prefixes, implicit and duplicate withdrawals as well as average and maximum AS-path length and edit distance and packet size.

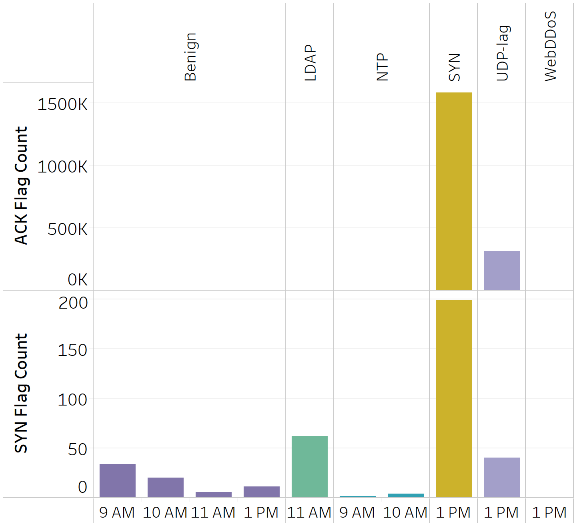

Fig. 3. CICDDoS2019, December 1, 2018: ACK (top) and SYN (bottom) flag counts. SYN attack degrades a network connection with a large number of seemingly legitimate TCP requests. The number of flags in the case of WebDDoS was minimal.

The DDoS2019 BGP dataset contains DDoS attacks targeting AWS (October 22, 2019) and banks in South Africa (October 23, 2019). The number of announced NLRI prefixes is shown in Fig. 4 (top). The DDoS attack that targeted AWS caused an eight-hour outage that affected the Amazon Route 53 cloud web service and left thousands of customers unable to access websites and applications [3]. The attack on October 22, 2019 occurred between 10:30 AM and 6:30 PM PDT and was persistent in San Francisco and intermittent in Boston, Chicago, and Dallas. On October 23, 2019, a wave of ransom driven DDoS attacks targeted the banking industry in South Africa that left Johannesburg's emergency call centers and e-services (including online banking and billing system) inaccessible to customers [8].

The DDoS2020 BGP dataset contains the largest volumetric DDoS attack of 2.3 Tbps that commenced on February 17, 2020 and caused three days of elevated security threat [1]. It was categorized as a Connectionless Lightweight Directory Access Protocol reflection attack that targeted Amazon cloud web services. This attack was 44 % larger than any AWS network volumetric event previously detected. The number of announced NLRI prefixes is shown in Fig. 4 (bottom).

IV. Experimental Procedure

The experiments are performed on Windows 10 64-bit Operating System and Intel Core i7-8650U CPU (1.9-2.11 GHz) using Python 3.8. PyTorch [4] is used to create the Bi-LSTM model.

A. *Data Processing: CIC-IDS Datasets*

The numbers of extracted features in CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019 are 84, 83, and 87, respectively [2]. We select 20 most important features from each dataset using the extra-trees ensemble method [9]. After converting categorical to numerical features, we apply min-max scaling to normalize features (between zero and one)

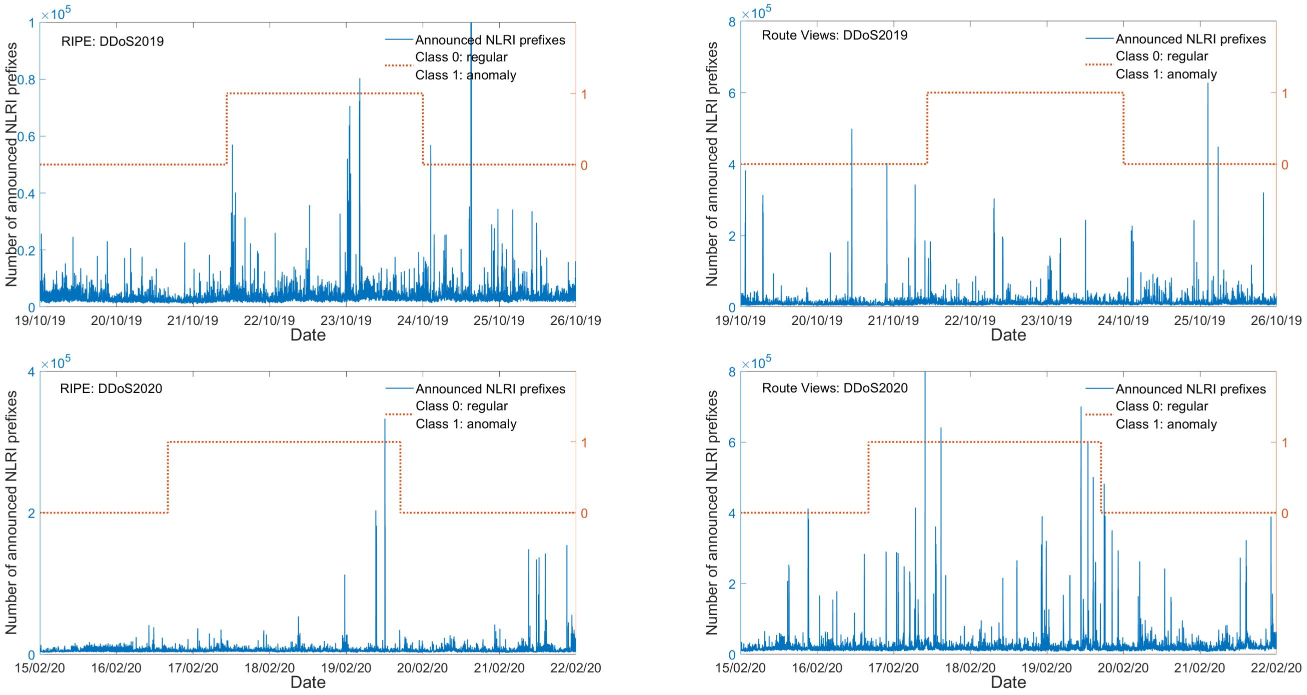

Fig. 4. BGP DDoS2019 (top) and DDoS2020 (bottom) datasets: Number of NLRI prefixes collected from RIPE (left) and Route Views (right) data collection sites. Shown are windows of anomalous (class 1) and regular (class 0) traffic.

to achieve proportional representation. The number of data points in training and test datasets used in the experiments is given in Table I. CICDDoS2019 dataset consists of multiple collections. Files are merged, randomly shuffled, and reduced to only include the first 10^6 points. We use naïve resampling technique to deal with the unbalanced CICDDoS2019 dataset. Random oversampling of minority (regular) class is applied and randomly selected samples are added to the dataset thus achieving balance between classes. The final dataset contains 50 % of regular data points.

TABLE I
CIC-IDS Datasets: Number of Data Points

Dataset	Class	Total	Training set	Test set
CICIDS2017	**Total**	**346,352**	**277,081**	**69,271**
Wednesday	Regular	219,984	175,855	44,129
July 5, 2017	Anomaly	126,368	101,226	25,142
CSE-CIC-IDS2018	**Total**	**525,288**	**419,430**	**104,858**
Thursday	Regular	497,973	398,349	99,624
February 15, 2018	Anomaly	26,315	21,081	5,234
CICDDoS2019	**Total**	**500,000**	**400,000**	**100,000**
Saturday	Regular	249,977	200,016	49,961
December 1, 2018	Anomaly	250,023	199,984	50,039

B. Data Processing: BGP Datasets

We create datasets used in experiments by considering known periods of the attacks (anomalous data points) as well as two days prior and two days after each attack (regular data points). Note that all data points within the window of an attack are considered anomalous. The Slammer, Nimda, Code Red I, and DDoS2020 datasets are partitioned using 80 % and 20 % of the total number of data points to create the training and test datasets, respectively. The DDoS2019 training and test datasets consist of 60 % and 40 % of the total number of data points, respectively. The number of data points in training and test datasets used in the experiments is given in Table II. The 20 most important features from each dataset are used to train the models.

TABLE II
BGP Datasets: Number of Data Points

Dataset	Class	Total	Training set	Test set
Slammer	**Total**	**7,200**	**5,760**	**1,440**
	Regular	6,331	5,058	1,273
	Anomaly	869	702	167
Nimda	**Total**	**8,609**	**6,887**	**1,722**
	Regular	7,308	5,841	1,467
	Anomaly	1,301	1,046	255
Code Red I	**Total**	**7,200**	**5,760**	**1,440**
	Regular	6,600	5,272	1,328
	Anomaly	600	488	112
DDoS2019	**Total**	**10,080**	**6,048**	**4,032**
	Regular	6,390	3,823	2,567
	Anomaly	3,690	2,225	1,465
DDoS2020	**Total**	**10,080**	**8,064**	**2,016**
	Regular	5,709	4,572	1,136
	Anomaly	4,371	3,492	880

C. Echo State Network Models

ESN models shown in Table III are developed by varying hyperparameters and using 10-fold cross validation [9] based on the time series split. Performance of ESNs is proportional to the number of reservoir nodes because they enable linear combination of the inputs to approximate the target labels. However, larger reservoirs are computationally expensive.

TABLE III

ESN MODELS AND HYPERPARAMETERS: DETERMINISTIC RESERVOIR WEIGHTS $\mathbf{W}$, SPECTRAL RADIUS $\rho(\mathbf{W})$, LEAKING RATE α, AND NUMBER OF RESERVOIR NODES N_z

	$\mathbf{W}$	$\rho(\mathbf{W})$	α	N_z
ESN$_1$	Random	0.9	0.2	10
ESN$_2$	Deterministic	0.9	0.2	10
ESN$_3$	Random	0.1	0.2	10
ESN$_4$	Random	0.9	1	10
ESN$_5$	Random	0.9	0.2	30

The input weight matrix $\mathbf{W}^{in}$ is randomly generated using binomial distribution with $k = 1$ trials and probability of success $p = 0.5$. The input weights scaling is set to 0.3. The randomly generated reservoir weights $\mathbf{W}$ are uniformly distributed between $[-0.5, 0.5]$ with matrix sparsity set to 75 %. A simple deterministically generated reservoir was shown to be appropriate for a variety of tasks and achieved performance comparable to the ESNs with randomly generated reservoirs [25]. Hence, we select a deterministic reservoir with identical connection weights equal to the value of spectral radius for one of the ESN models.

D. Bidirectional LSTM Model

Bi-LSTM neural network is a variant of LSTM with two hidden layers in opposite directions connected to the same output [26]. It improves sequence classification tasks due to its ability to utilize information based on past (backward) and future (forward) direction states. Forward and backward cell states are independent and, therefore, no delays are introduced when using future information.

The evaluated Bi-LSTM contains 20 input nodes, 16 output nodes, dropout rate $= 0.5$, batch size $= 10$, and $ReLU$ activation function. The fully-connected layer has 32 input and 2 output nodes that are passed to the $softmax$ module. The best performance was achieved with learning rate of 0.001 among the $0.1, 0.01, 0.001$ values. The optimization algorithm "Adam" is used to train the Bi-LSTM models using 10 epochs.

V. PERFORMANCE OF ECHO STATE NETWORK MODELS

We use accuracy, F-Score, and false alarm rate (FAR) as performance metrics based on confusion matrix elements true positive (TP), false positive (FP), true negative (TN), and false negative (FN):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{FAR} = \frac{FP}{TN + FP}. \tag{8}$$

TABLE IV

PERFORMANCE OF ESN AND BI-LSTM MODELS USING CIC-IDS AND BGP DATASETS

	CICIDS2017			**Slammer**		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN$_1$	0.927	0.907	0.106	0.907	0.699	0.080
ESN$_2$	0.958	0.945	0.058	0.908	0.710	0.083
ESN$_3$	0.915	0.893	0.120	0.930	0.726	0.036
ESN$_4$	0.919	0.899	0.120	0.927	0.712	0.036
ESN$_5$	0.962	0.950	0.053	0.900	0.699	0.095
Bi-LSTM	0.995	0.994	0.002	0.958	0.827	0.024
	Training time (s)					
ESN$_5$	988			8		
Bi-LSTM	2,200			34		
	CIC-CSE-IDS2018			**Nimda**		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN$_1$	0.983	0.854	0.017	0.805	0.502	0.166
ESN$_2$	0.980	0.828	0.020	0.821	0.470	0.130
ESN$_3$	0.961	0.679	0.032	0.843	0.167	0.024
ESN$_4$	0.979	0.824	0.021	0.841	0.122	0.021
ESN$_5$	0.997	0.973	0.003	0.818	0.516	0.150
Bi-LSTM	0.996	0.962	0.004	0.863	0.375	0.029
	Training time (s)					
ESN$_5$	2,335			7		
Bi-LSTM	3,417			41		
	CICDDoS2019			**Code Red I**		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN$_1$	0.994	0.994	0.012	0.910	0.432	0.040
ESN$_2$	0.991	0.992	0.016	0.919	0.424	0.027
ESN$_3$	0.927	0.932	0.146	0.913	0.046	0.002
ESN$_4$	0.981	0.999	0.000	0.901	0.536	0.075
ESN$_5$	0.999	0.999	0.001	0.910	0.547	0.062
Bi-LSTM	1.000	1.000	0.000	0.929	0.491	0.021
	Training time (s)					
ESN$_5$	1,690			6		
Bi-LSTM	2,619			37		
	DDoS2019, RIPE			**DDoS2019, Route Views**		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN$_1$	0.571	0.502	0.465	0.613	0.433	0.259
ESN$_2$	0.579	0.558	0.527	0.611	0.551	0.406
ESN$_3$	0.481	0.522	0.702	0.615	0.261	0.130
ESN$_4$	0.525	0.505	1.000	0.624	0.193	0.084
ESN$_5$	0.677	0.617	0.371	0.618	0.540	0.373
Bi-LSTM	0.388	0.478	0.837	0.654	0.791	1.000
	Training time (s)					
ESN$_5$	12			6		
Bi-LSTM	111			99		
	DDoS2020, RIPE			**DDoS2020, Route Views**		
	Acc.	F-Score	FAR	Acc.	F-Score	FAR
ESN$_1$	0.439	0.610	0.988	0.477	0.609	0.877
ESN$_2$	0.437	0.606	0.994	0.577	0.610	0.565
ESN$_3$	0.437	0.607	0.998	0.437	0.603	0.982
ESN$_4$	0.436	0.607	1.000	0.441	0.604	0.971
ESN$_5$	0.453	0.610	0.955	0.595	0.621	0.536
Bi-LSTM	0.346	0.514	1.000	0.760	0.864	1.000
	Training time (s)					
ESN$_5$	9			11		
Bi-LSTM	107			101		

Performance of the ESN models using CIC-IDS and BGP datasets is given in Table IV. The ESN$_5$ model using CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019 datasets leads to the best accuracy, F-Score, and FAR. Increasing the number of reservoir nodes N_z enhances the performance of ESN models: The ESN$_5$ model with 30 reservoir nodes shows better performance than ESN$_1$ model with 10 reservoir nodes. Reducing the radius of the reservoir degrades the performance as illustrated by performance of ESN$_3$ model with low spectral radius.

The ESN$_3$ and ESN$_4$ models, however, exhibit better performance using Slammer dataset. The ESN$_2$ and ESN$_5$ models achieve better performance using Nimda and Code Red I datasets. When using DDoS2019 and DDoS2020 datasets, the ESN models are unable to adequately classify anomalies because patterns of anomalous and regular traffic are similar. The models classify almost all data points as regular leading to low recall and F-Score. Data points in BGP datasets are labeled based on reported periods of anomalous events. However, the selected anomaly windows may also contain regular BGP traffic while regular windows may contain diverse BGP anomalies as in the case of spikes in the number of BGP announcements and announced NLRI prefixes observed on February, 22, 2020.

The employed datasets influence performance of ESN models. The ESN models evaluated using CIC-IDS datasets exhibit better performance than with BGP datasets. Note that CIC-IDS datasets are synthetically generated and contain records of various application layer protocols. In contrast, BGP datasets DDoS2019 and DDoS2020 capture records of BGP protocol from deployed networks. The CIC-IDS datasets that we consider consist of approximately 0.5×10^6 data points while the BGP datasets contain 10^4 data points. Hence, training the ESN models using larger datasets may have also contributed to improved performance.

Compared to the Bi-LSTM model, the ESN$_5$ model shows better performance when evaluated using CSE-CIC-IDS2018 as well as BGP DDoS2019 and DDoS2020 datasets collected by RIPE and Route Views. As shown in Table IV, Bi-LSTM models offer comparable or better results using Slammer, Nimda, and Code Red I datasets. Training the ESN models requires shorter training time because they do not employ backpropagation used by the Bi-LSTM models.

VI. Conclusions

We evaluated performance of ESN and Bi-LSTM models to detect various DoS and DDoS attacks by using CIC-IDS synthetic datasets as well as RIPE and Route Views BGP datasets collected from deployed networks. A number of ESN models was designed by varying hyperparameters of the reservoir network such as the type of generated reservoir weights, spectral radius, leaking rate, and number of nodes. Increasing the number of reservoir nodes and the radius of the reservoir enhanced the model performance. The ESN and Bi-LSTM models evaluated in this paper demonstrated comparable accuracy, F-Score, and FAR while ESN models required shorter training time. Even though performance of the classifiers was influenced by the employed datasets, experimental results illustrated that ESNs may be used to successfully detect network anomalies.

References

[1] AWS Shield Threat Landscape Report - Q1 2020. [Online]. Available: https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf. Accessed: Aug. 31, 2021.

[2] Canadian Institute for Cybersecurity Datasets. [Online]. Available: https://www.unb.ca/cic/datasets/index.html Accessed: Aug. 31, 2021.

[3] Learning From the Amazon Web Services (AWS) DDoS Attack. [Online]. Available: https://www.corero.com/blog/learning-from-the-amazon-web-services-aws-ddos-attack/. Accessed: Aug. 31, 2021.

[4] PyTorch. [Online]. Available: https://pytorch.org/docs/stable/nn.html. Accessed: Aug. 31, 2021.

[5] RIPE NCC: RIPE Network Coordination Center. [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data. Accessed: Aug. 31, 2021.

[6] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, *IETF*, Mar. 1995. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1771.txt.

[7] University of Oregon Route Views project. [Online]. Available: http://www.routeviews.org. Accessed: Aug. 31, 2021.

[8] South African Banks Resilient in the Face of Latest DDoS Attacks. [Online]. Available: https://www.sabric.co.za/media-and-news/press-releases/south-african-banks-resilient-in-the-face-of-latest-ddos-attacks/. Accessed: Aug. 31, 2021.

[9] Scikit-learn: machine learning in Python. [Online]. Available: https://scikit-learn.org/ Accessed: Aug. 31, 2021.

[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] M. Chang, A. Terzis, and P. Bonnet, "Mote-based online anomaly detection using echo state networks," in *Lecture Notes in Computer Science: Proc. Distrib. Comput. in Sensor Syst.,* Berlin: Springer, Berlin, 2009, vol. 5516, pp. 72–86.

[13] 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.

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

[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] A. L. Gonzalez Rios, Z. Li, K. Bekshentayeva, and Lj. Trajkovic, "Detection of denial of service attacks in communication networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seville, Spain, Oct. 2020.

[17] H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, Dec. 2007.

[18] 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.

[19] 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.

[20] Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, "Detecting Internet worms, ransomware, and blackouts using recurrent neural networks," in *IEEE Int. Conf. Syst., Man, and Cybern.*, Toronto, Canada, Oct. 2020, pp. 2165–2172 (virtual).

[21] Z. Li, A. L. Gonzalez Rios, and Lj. Trajkovic, "Machine learning for detecting anomalies and intrusions in communication networks," *IEEE J. Select. Areas Commun.*, vol. 39, no. 7, pp. 2254–2264, Jul. 2021.

[22] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI. Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, Nov. 2012.

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

[24] Z. Ran, D. Zheng, Y. Lai, and L. Tian, "Applying stack bidirectional LSTM model to intrusion detection," *Comput., Mater. & Continua*, vol. 65, no. 1, pp. 309–320, May 2020.

[25] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Trans. on Neural Netw.*, vol. 22, no. 1, pp. 131–144, Jan. 2011.

[26] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.