Network Intrusion Detection Using Various Deep Learning Approaches

Matt Canute, Young-min Kim, Adriena Wong, Donggu Lee, and Kamila Bekshentayeva

Simon Fraser University

PROBLEM

Cyber attacks are continuously evolving and becoming more sophisticated and difficult to detect.

QUESTIONS

How do deep learning models detect network intrusions?

Can they be bypassed by adjusted attacks?

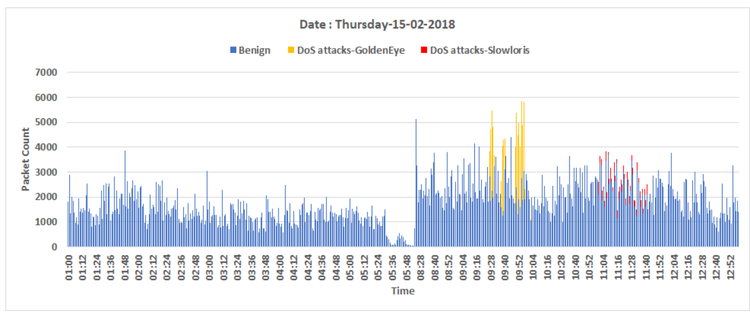
If so, what approaches are less easily fooled?

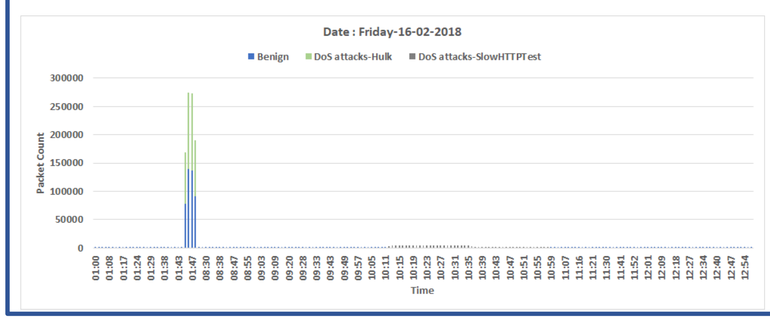
GOAL

To experiment with various adversarial attacks, custom intrusion sets, and various neural network architectures, employing Communications Security Establishment (CSE) & the Canadian Institute of Cybersecurity (CIC) intrusion data set of 2018 that reflects current network trends.

DATA SET

75 Traffic Features of 6.1 Million Flows in total

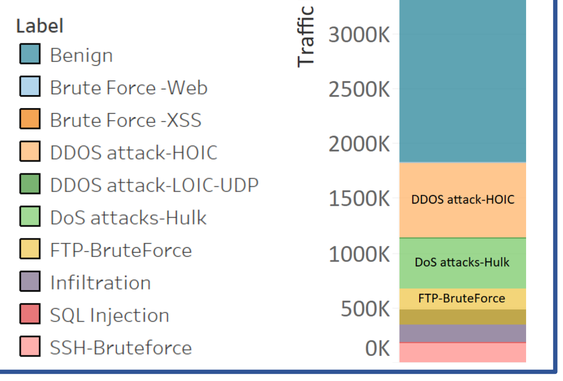




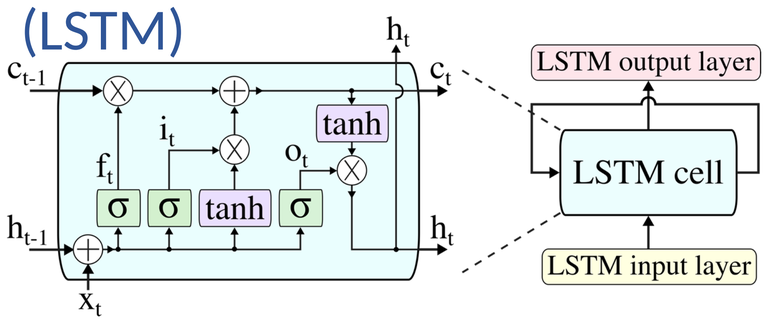
CSE-CIC-IDS2018

Testbed: Attacker-network with 50 terminals, and victim-network, implemented as a Local Area Network (LAN) with 420 terminals and 30 servers divided into 5 subnets [1,2].

Benign data: background traffic based on user non-malicious behavior when employing: HTTP, HTTPS, SMTP, POP3, IMAP, SSH, and FTP.



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:

$$\mathbf{f_t} = \sigma\left(\mathbf{W_{if}x_t} + \mathbf{b_{if}} + \mathbf{U_{hf}h_{t-1}} + \mathbf{b_{hf}}\right)\mathbf{i_t} = \sigma\left(\mathbf{W_{ii}x_t} + \mathbf{b_{ii}} + \mathbf{U_{hi}h_{t-1}} + \mathbf{b_{hi}}\right)$$
$$\mathbf{o_t} = \sigma(\mathbf{W_{io}x_t} + \mathbf{b_{io}} + \mathbf{U_{ho}h_{t-1}} + \mathbf{b_{ho}})$$

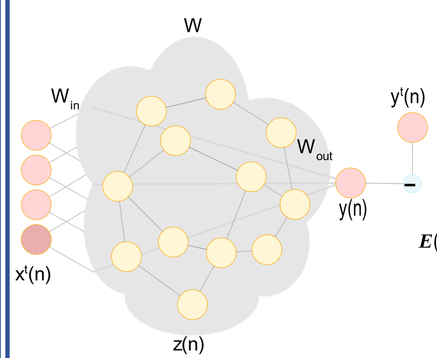
The cell state c_t is calculated as:

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

The output of the LSTM cell is:

$$\mathbf{h_t} = \mathbf{o_t}\tanh(\mathbf{c_t})$$

Echo State Networks (ESN)



Update equations:

$$\tilde{\boldsymbol{z}}(\boldsymbol{n}) = \tanh\left(\boldsymbol{W}^{in}[\mathbf{1}; \boldsymbol{x}(\boldsymbol{n})] + \boldsymbol{W}\boldsymbol{z}(\boldsymbol{n}-\mathbf{1})\right)$$
$$\boldsymbol{z}(\boldsymbol{n}) = (\mathbf{1}-\boldsymbol{\alpha})\boldsymbol{z}(\boldsymbol{n}-\mathbf{1}) + \boldsymbol{\alpha}\tilde{\boldsymbol{z}}(\boldsymbol{n})$$

Linear output layer::

$$\boldsymbol{y}(\boldsymbol{n}) = \boldsymbol{W}\boldsymbol{out}[\mathbf{1}; \boldsymbol{x}(\boldsymbol{n}); \boldsymbol{z}(\boldsymbol{n})]$$

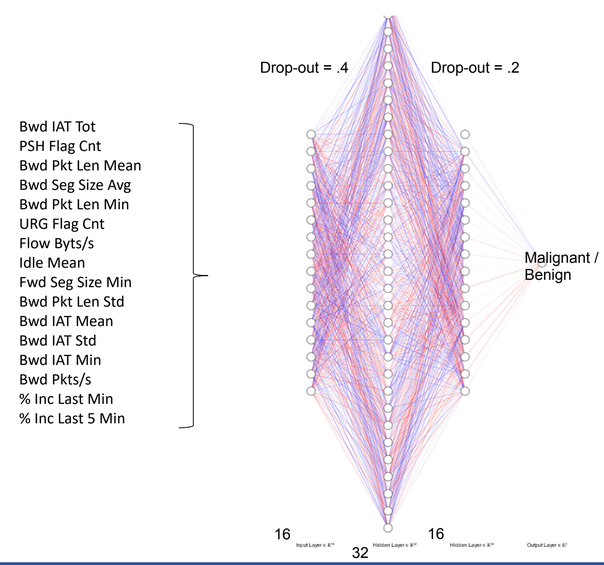
$$\boldsymbol{E}(\boldsymbol{y}, \boldsymbol{y}^t) = \frac{\mathbf{1}}{\boldsymbol{N_y}} \sum_{i=1}^{N_y} \sqrt{\frac{\mathbf{1}}{\boldsymbol{T}} \sum_{n=1}^{T} \left(\boldsymbol{y_i}(\boldsymbol{n}) - \boldsymbol{y_i^t}(\boldsymbol{n})\right)^2}$$

Steps associated with applying ESN [3, 4]:

1. Generation of random reservoir (reservoir size T < N^x +N^z+1) with the following parameters: W^{in}, W, α ϵ [0,1);
2. Obtaining the reservoir activation states z(n) by using the training input x(n);
3. Obtaining the output weights W^{out}, while minimizing the error;
4. Applying collected output weights W^{out} with new input x(n) to compute y(n).

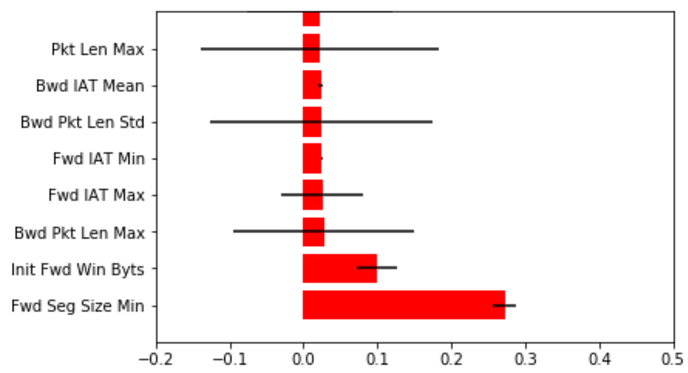
Values used:
Spectral radius: 0.6
Leaking rate: 0.6
Connectivity: 0.25
Input scaling: 0.1
Noise level: 0.01
Number of features used: 10-20
Days: Thursday: random sample of 50% of data; Friday: random sample of 75% of data

Feed-Forward Neural Networks (FFNN)

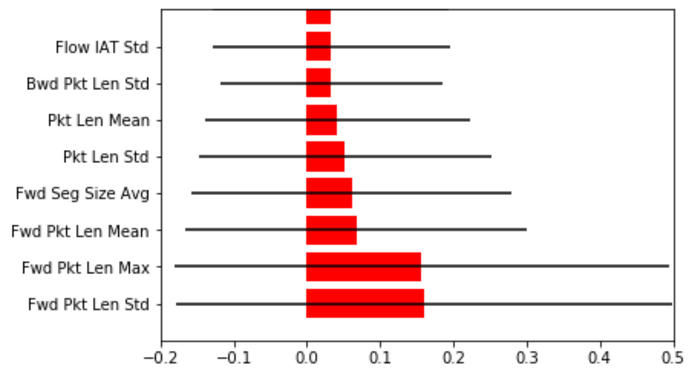


EXPERIMENTS (Feature selection and adversarial landscape)

Importance of Traffic Features for detecting GoldenEye/Slowloris Attacks



Importance of Traffic Features for detecting Hulk/Slow HTTP Attacks



Feature Engineering/Selection

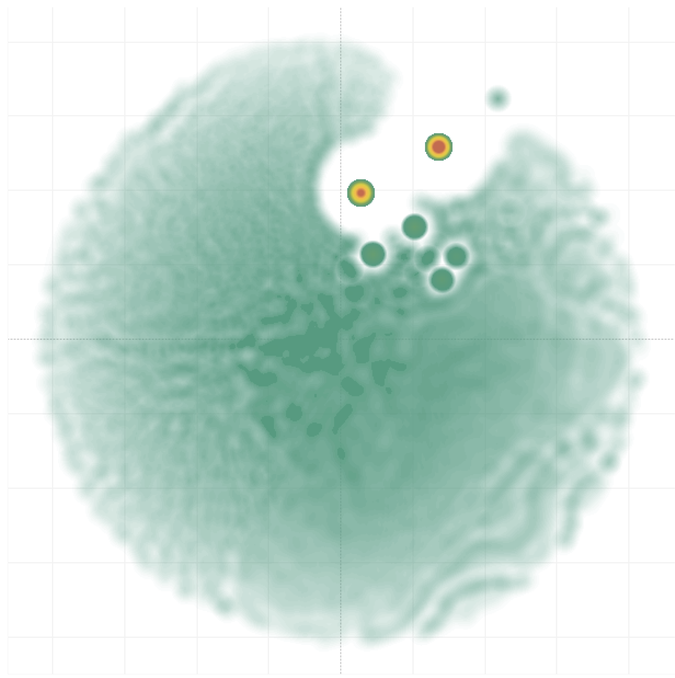
Trained Models on 2018 attacks

LSTM

ESN

FFNN

Additional Test on Modified 2019 Attack



T-SNE of a Hulk DOS Attack versus the 99% Accurate FFNN Model

RESULTS

LSTM

LSTM Test Set Hulk/Slow HTTP Attack: 99.5%

		Actual: Benign	Actual: Attack
Predicted	Benign	88,450	
	Attack	947	120,318

Test Set GoldenEye/Slowloris Attacks:95.0%

		Actual: Benign	Actual: Attack
Predicted	Benign	199,226	10,489
	Attack		

LSTM New 2019 Hulk Attack: 97.1%

		Actual: Benign	Actual: Attack
Predicted	Benign	9	
	Attack	249	8,450

ESN

ESN Test Set Hulk/Slow HTTP Attack: 98.9%

		Actual: Benign	Actual: Attack
Predicted	Benign	82,573	1,411
	Attack	847	111,777

Test Set GoldenEye/Slowloris Attacks: 93.9%

		Actual: Benign	Actual: Attack
Predicted	Benign	122,927	6,497
	Attack	1,561	87

ESN New 2019 Hulk Attack: 97.2%

		Actual: Benign	Actual: Attack
Predicted	Benign	11	0
	Attack	247	8,450

FFNN

FFNN Test Set Hulk/Slow HTTP Attack: 99.8%

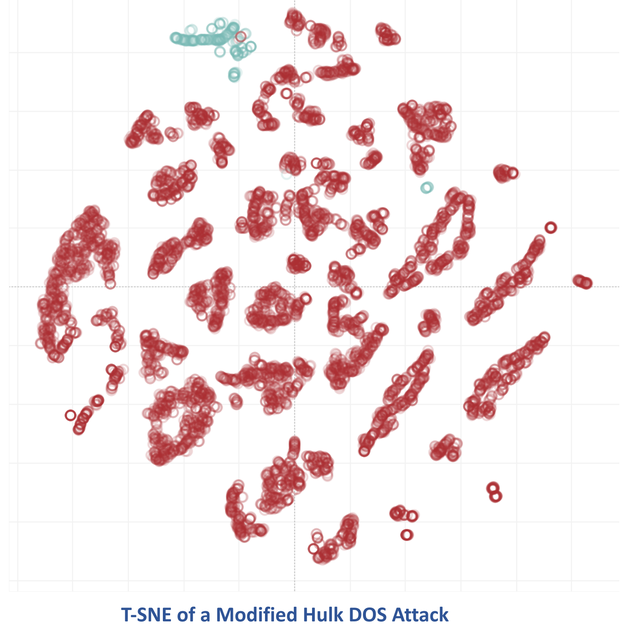
		Actual: Benign	Actual: Attack
Predicted	Benign	89,265	337
	Attack	-	120,113

Test Set GoldenEye/Slowloris Attacks:99.7%

		Actual: Benign	Actual: Attack
Predicted	Benign	199,005	145
	Attack	572	9,993

FFNN New 2019 Hulk Attack: 3.2%

		Actual: Benign	Actual: Attack
Predicted	Benign	258	8,432
	Attack	-	18



T-SNE of a Modified Hulk DOS Attack

The red marks are all the examples that the same FFNN Model pictured above 'missed'

References:

[1] Intrusion Detection Evaluation Dataset (CICIDS2017) [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. Accessed: Oct. 28, 2019.
[2] CICFlowMeter [Online]. Available: http://netflowmeter.ca/\\netflowmeter.html. Accessed: Oct. 28, 2019.
[3] M. Lukoševičius, "A Practical Guide to Applying Echo State Networks," in *Neural Networks*: Tricks of the Trade. Springer, 2012, pp.659-686.
[4] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks, " Tech. Rep. GMD Rep. 148, German Nat. Res. Center for Inf. Technol., 2001.
[5] Z. Li, P. Batta, and Lj. Trajkovic, "Comparison of machine learning algorithms for detection of network intrusions," in Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC 2018), Miyazaki, Japan, Oct. 2018, pp. 4248-4253.