Spiking Neural Networks with Single-Spike Temporal-Coded Neurons for Network Intrusion Detection

Shibo Zhou and Xiaohua Li
Binghamton University, State University of New York, USA

Abstract

Considering a general class of single-spike temporal-coded integrate-and-fire neurons, we analyze the input-output expressions of both leaky and nonleaky neurons. We show that SNNs built with leaky neurons suffer from the overly-nonlinear and overly-complex input-output response, which is the major reason for their difficult training and low performance. This is more fundamental than the commonly believed problem of nondifferentiable spikes. To support this claim, we show that SNNs built with nonleaky neurons can have a less-complex and less-nonlinear input-output response. They can be easily trained and can have superior performance, which is demonstrated by experimenting with the SNNs over two popular network intrusion detection datasets, i.e., the NSL-KDD and the AWID datasets.

Objectives

- We analyze input-output response of two general types of SNN neurons to show that the commonly used leaky neurons have too complex and too nonlinear input-output response and are thus hard to train.
- We show that SNNs built with nonleaky neurons can have much less complex and much less nonlinear input-output response.
- We train the proposed SNNs over two popular network intrusion detection datasets NSL-KDD and AWID. New benchmark results are obtained.

Methods

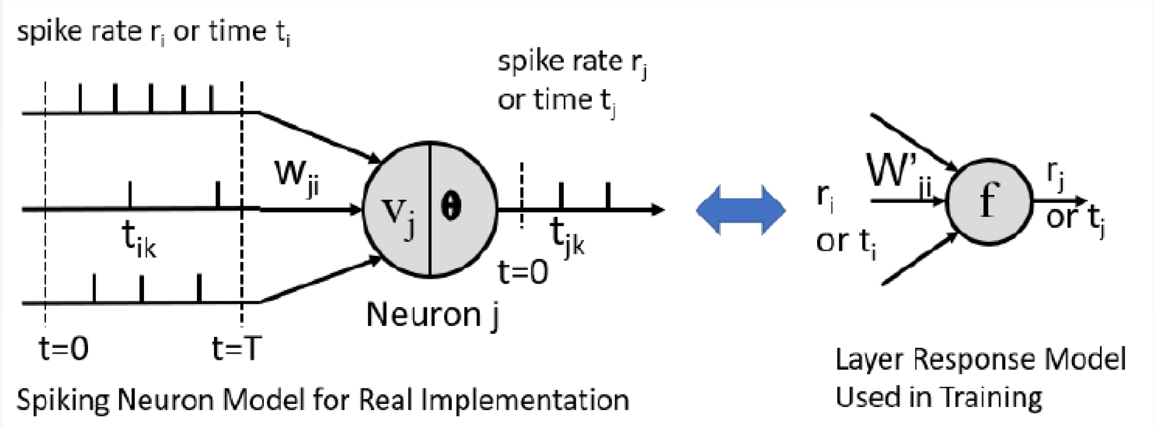
We develop SNN models based on the nonleaky neurons and consider neurons that use spiking time to encode information. Each neuron emits a single spike only for energy efficiency.

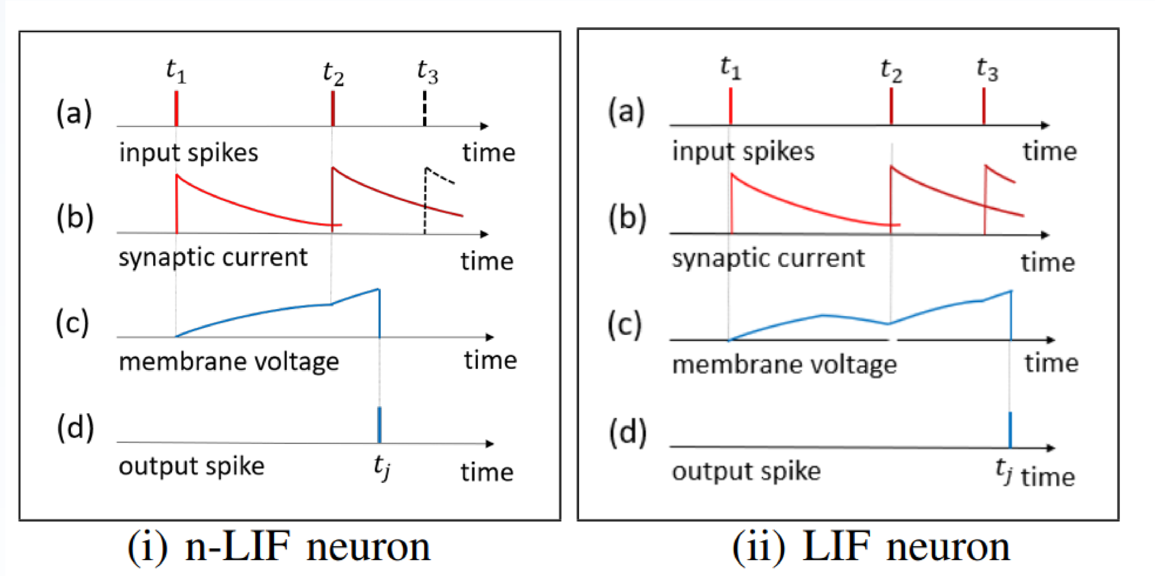
For the non-leaky integrate-and-fire (n-LIF) neuron:

$$\frac{dv_j(t)}{dt} = \sum_i w_{ji} g(t - t_i)$$

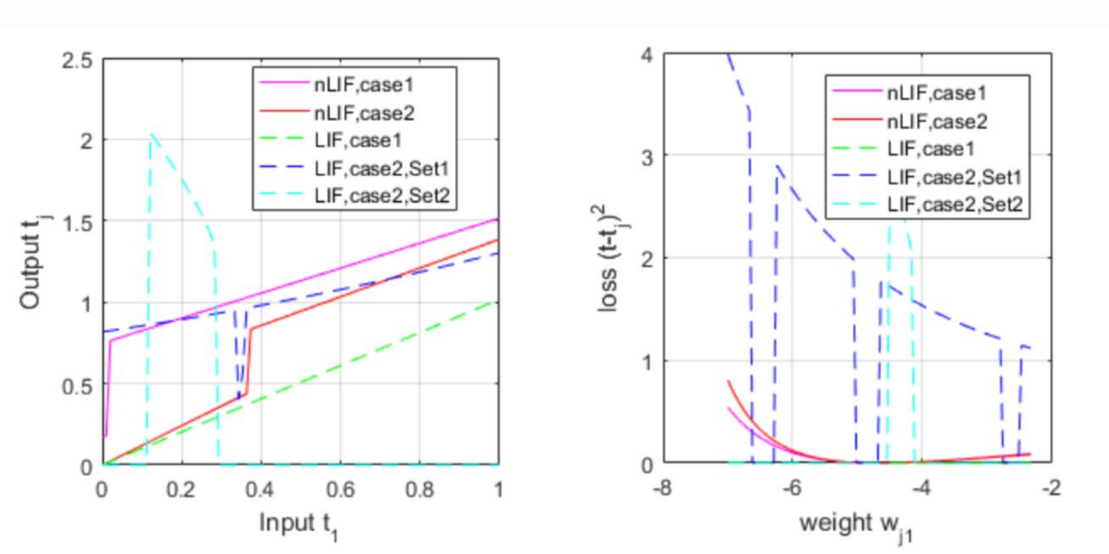
For the leaky integrate-and-fire (LIF) neuron:

$$\frac{dv_j(t)}{dt} + b v_j(t) = \sum_i w_{ji} g(t - t_i)$$





(i) n-LIF neuron (ii) LIF neuron



SNN Algorithm

Algorithm 1 Forward pass of a SNN neuron based on (11)

Input: $\mathbf{z} = [z_1, \cdots, z_N]$: input spiking time vector
Input: $\mathbf{w} = [w_1, \cdots, w_N]$: weight vector
Output: z_{out}: output spiking time
$\mathbf{i} \leftarrow argsort(\mathbf{z})$: ascending order index
$\mathbf{z}_{sorted} \leftarrow \mathbf{z}[\mathbf{i}]$: sorted input vector
$\mathbf{w}_{sorted} \leftarrow \mathbf{w}[\mathbf{i}]$: sorted weight vector
$\mathbf{z}_{cumsum} \leftarrow cumsum(\mathbf{z}_{sorted} * \mathbf{w}_{sorted})$: $\sum_{i \in C} w_{ji} z_i$
$\mathbf{w}_{cumsum} \leftarrow cumsum(\mathbf{w}_{sorted}) - v_0/\tau$: $\sum_{\ell} w_{j\ell} - v_0/\tau$
$\mathbf{z}_{candidate} \leftarrow \mathbf{z}_{cumsum}/\mathbf{w}_{cumsum}$: element-wise division
$\mathbf{z}_{shifted} \leftarrow [\mathbf{z}_{sorted}[2:], Inf]$: next input spiking time
$\mathbf{c} \leftarrow (\mathbf{z}_{candidate} > \mathbf{z}_{sorted}) \& (\mathbf{z}_{candidate} <= \mathbf{z}_{shifted})$
$k \leftarrow where(\mathbf{c} == True)[1]$: index of the first True element
return $z_{out} \leftarrow \mathbf{z}_{candidate}[k]$: output spiking time

Datasets

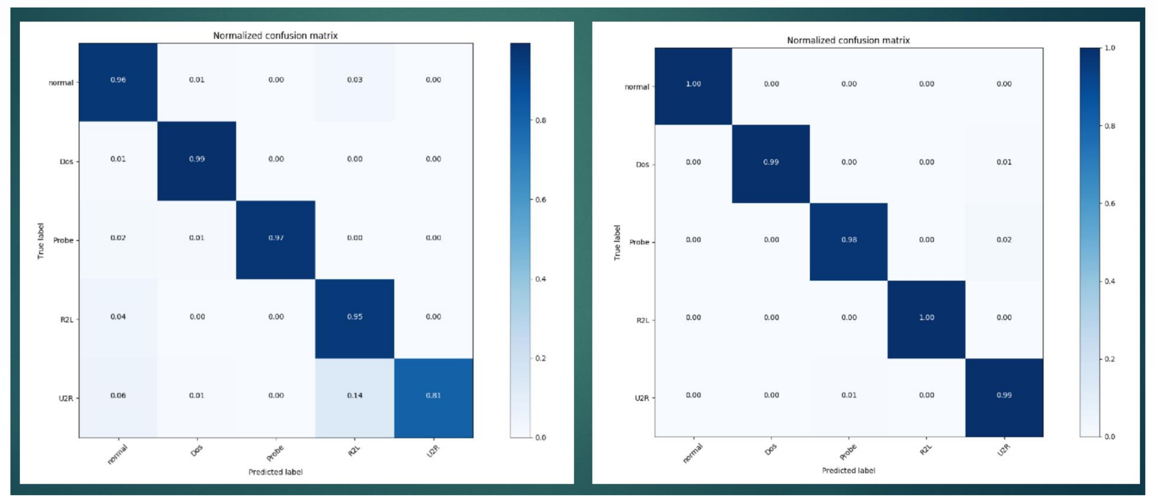
Two major datasets for network intrusion detection research:

NSL-KDD, AWID

Results-NSLKDD

Our SNNs have the best performance (highest accuracy) than DNN/CNN, RL, and traditional machine learning methods

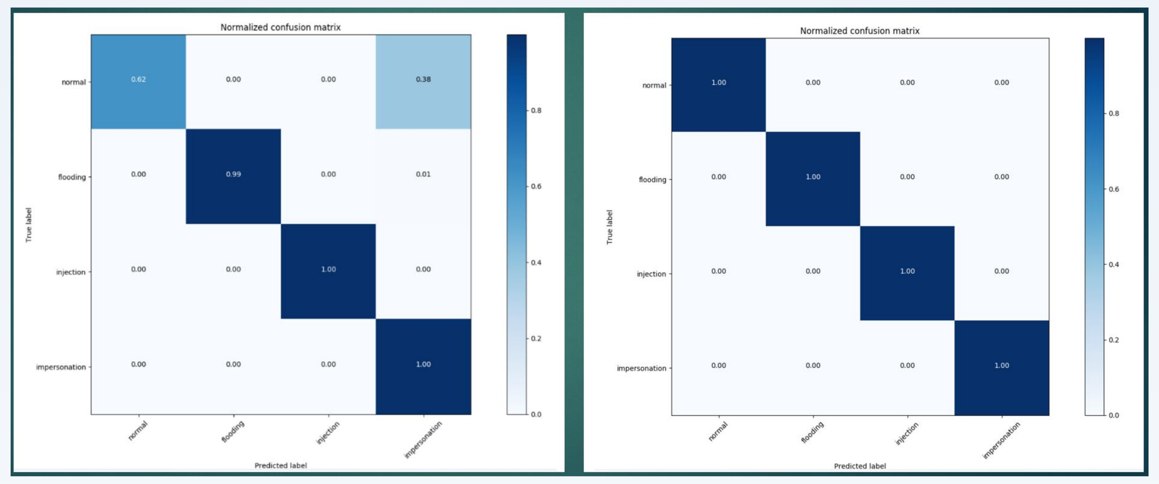
	Accuracy	F1	Precision	Recall
Logistic Regression	0.7068	0.6807	0.8955	0.5491
SVM	0.8799	0.8927	0.9081	0.8779
KNN	0.7808	0.7769	0.9233	0.6706
Random Forest	0.7472	0.7211	0.9688	0.5743
radient Tree Boosting	0.7761	0.7612	0.9690	0.6267
Naïve Bayes	0.8019	0.7967	0.9583	0.6818
AdaBoost	0.7606	0.7403	0.9583	0.5992
Neural Network	0.7966	0.7881	0.9679	0.6647
CNN-1D	0.7875	0.7633	0.8094	0.7875
Reinforcement Learn	0.8978	0.9120	0.8944	0.9303
Our DNN	0.8834	0.8860	0.8936	0.8834
Our CNN-1D	0.9564	0.9561	0.9565	0.9564
SNN (Original)	0.9717	0.9718	0.9721	0.9717
SNN (Resampled)	**0.9931**	**0.9931**	**0.9931**	**0.9931**



Results-AWID

Our SNNs have the best performance (highest accuracy) than DNN/CNN, RL, and traditional machine learning methods.

	Accuracy	F1	Precision	Recall
AdaBoost	0.9220	0.8850	0.8500	0.9220
Decision Tree	0.9620	0.9480	0.9620	0.9630
Naïve Bayes	0.9055	0.9090	0.9170	0.9060
Frequency Tabel	0.9457	0.9220	0.9000	0.9460
Random Forest	0.9582	0.9440	0.9590	0.9580
Neural Network	0.9470	0.9256	0.9174	0.9473
Reinforcement Learn	0.9570	0.9394	0.9235	0.9570
Our DNN	0.9585	0.9624	0.9715	0.9585
Our CNN-1D	0.9528	0.9351	0.9504	0.9528
SNN (Original)	0.9898	0.9893	0.9895	0.9898
SNN (Resampled)	**0.9984**	**0.9985**	**0.9985**	**0.9984**



Conclusions

We develop single-spike temporal-coded SNNs that can be easily trained with competitive performance as conventional DNNs. We analyze systematically the input-output expressions of single-spike temporal-coded leaky and nonleaky neurons. We also show that the SNNs with n-LIF neurons outperform a list of existing methods including the DNN-based methods.

References

M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, “Application of deep reinforcement learning to intrusion detection for supervised problems,” *Expert Systems with Applications,* Vol. 141, P.