NF-GNN: Network Flow Graph Neural Networks for Network Traffic Based Malware Detection and Classification – Supplementary Material –

Anonymous Authors*

Anonymized

A Hyper-Parameter Optimization

For the sake of a fair comparison, hyper-parameters of all models are optimized by a grid search based on performance on a separate validation set. For each algorithm, we consider a set of possible values for each of its adjustable hyper-parameters. While some common choices can be found in the literature, for the remaining hyper-parameters we determine potential values which seem most promising within the available computational budget. The considered hyper-parameter values for all supervised models can be found in Table 1, the values considered for the unsupervised models are provided in Table 2.

B Influence of the Number of Message Passing Layers

To further evaluate the influence of the number of layers on the performance of our model, we compare different choices for the supervised classification tasks in Figure 1. We can observe that modeling 2-hop interactions between endpoints can boost performance on the binary prediction tasks, while direct interactions are more important for the remaining two tasks. In general, performance remains rather stable for different numbers of layers.

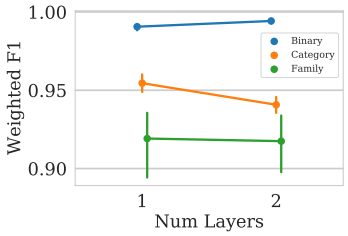

Fig. 1. Performance of our model on supervised tasks with different numbers of layers.

* Anonymized

Algorithm	**Parameter**	**Values**
Support Vector Machine (SVM)	C	$[2^{-7}, 2^{-6}, \ldots, 2^{7}]$
	γ (RBF-kernel)	$[2^{-7}, 2^{-6}, \ldots, 2^{7}]$
k-Nearest Neighbor Classifier (KNN)	num. neighbors	$[1, 2, 3, 5, 8, 13, 21]$
Decision Tree (DT)	max. depth	$[2, 5, 10, None]$
	max. features	$[sqrt, None]$
Random Forest (RF)	num. estimators	$[10, 100, 1000]$
	criterion	$[entropy, gini]$
	max. features	$[sqrt, None]$
Adaboost (ADA)	num. estimators	$[10, 100, 1000]$
	learning rate	$[1e-3, 1e-2, 1e-1, 1]$
Multi-Layer Perceptron (MLP)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	L2-reg.	$[0, 1e-1, 1e-2, 1e-3, 1e-4]$
NF-GNN-CLF (ours)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	learning rate	$[1e-3, 1e-2]$
	dropout prob.	$[0, 0.2, 0.4, 0.6]$
	pool	$[mean, add, max]$

Table 1. Hyper-parameter values used in grid search for supervised algorithms.

Algorithm	**Parameter**	**Values**
One-class SVM (OC-SVM)	ν	$[1e-2, 1e-1]$
	γ	$[2^{-10}, 2^{-9}, \ldots, 2^{10}]$
Local Outlier Factor (LOF)	num. neighbors	$[1, 2, 3, 5, 8, 13, 21]$
Kernel Density Estimation (KDE)	bandwidth	$[2^{0.5}, 2, \ldots, 2^5]$
Isolation Forest (IF)	num. estimators	$[10, 100, 1000]$
	max. features	$[256, None]$
Autoencoder (MLP-AE)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	L2-reg.	$[0, 1e-1, 1e-2, 1e-3, 1e-4]$
One-class MLP (MLP-OC)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	L2-reg.	$[0, 1e-1, 1e-2, 1e-3, 1e-4]$
NF-GNN-AE (ours)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	learning rate	$[1e-3, 1e-2]$
	dropout prob.	$[0, 0.2, 0.4, 0.6]$
	pool	$[mean, add, max]$
NF-GNN-OC (ours)	num. layers	$[1, 2]$
	num. hidden	$[16, 32, 64, 128]$
	learning rate	$[1e-3, 1e-2]$
	dropout prob.	$[0, 0.2, 0.4, 0.6]$
	pool	$[mean, add, max]$

Table 2. Hyper-parameter values used in grid search for unsupervised algorithms.