

# Evaluating Label Flipping Attack in Deep Learning-Based NIDS

Hesamodin Mohammadian<sup>1</sup><sup>a</sup>, Arash Habibi Lashkari<sup>2</sup><sup>b</sup> and Ali A. Ghorbani<sup>1</sup>

<sup>1</sup>Canadian Institute for Cybersecurity, University of New Brunswick, Fredericton, New Brunswick, Canada

<sup>2</sup>School of Information Technology, York University, Toronto, Ontario, Canada

**Keywords:** Network Intrusion Detection, Deep Learning, Poisoning Attack, Label Flipping.

**Abstract:** Network intrusion detection systems are one of the key elements of any cybersecurity defensive system. Since these systems require processing a high volume of data, using deep learning models is a suitable approach for solving these problems. But, deep learning models are vulnerable to several attacks, including evasion attacks and poisoning attacks. The network security domain lacks the evaluation of poisoning attacks against NIDS. In this paper, we evaluate the label-flipping attack using two well-known datasets. We perform our experiments with different amounts of flipped labels from 10% to 70% of the samples in the datasets. Also, different ratios of malicious to benign samples are used in the experiments to explore the effect of datasets' characteristics. The results show that the label-flipping attack decreases the model's performance significantly. The accuracy for both datasets drops from 97% to 29% when 70% of the labels are flipped. Also, results show that using datasets with different ratios does not significantly affect the attack's performance.

## 1 INTRODUCTION


Machine Learning has been extensively used in automated tasks and decision-making problems. There has been tremendous growth and dependence on using ML applications in national critical infrastructures and critical areas such as medicine and healthcare, computer security, spam, malware detection, autonomous driving vehicles, unmanned autonomous systems, and homeland security (Duddu, 2018). Deep learning has shown promising results in machine learning tasks in recent years. But, Recent studies show that machine learning, specifically deep learning models, are highly vulnerable to adversarial examples either at training or at test time (Biggio and Roli, 2018).


There are different kinds of attacks against deep learning models both in the training and test time, such as membership inference attack (Shokri et al., 2017; Hu et al., 2022), model inversion attack (Fredrikson et al., 2015), evasion attack (Szegedy et al., 2013), and poisoning attack (Tian et al., 2022). In a membership inference attack, the attacker aims to break the model's privacy and determine if a specific record is part of the model's training set. These attacks are significant when working with private data

such as health or financial-related information of the people (Truex et al., 2018; Choquette-Choo et al., 2021). The model inversion attack is close to the membership inference attack and aims to extract sensitive features of the training samples (Zhang et al., 2020). An evasion attack happens during the test phase. It aims to fool the trained deep learning model and bypass the detection by crafting adversarial examples by making small changes to the original samples of the dataset (Goodfellow et al., 2014). One of the first attempts to make this attack was in (Dalvi et al., 2004), where they studied this problem in spam filtering. Attackers can also affect the model's performance during the training phase, such as poisoning attacks where they try to inject poisoned data into the training set (Wang et al., 2022).

In recent years deep learning has shown its potential in the security area, such as malware detection and intrusion detection systems (NIDS). A NIDS aims to distinguish between benign and malicious behaviors inside a network (Buczak and Guven, 2015). With the rapid growth of network traffic, interest in using deep learning-based anomaly detection methods has increased. These techniques can provide more efficient and flexible approaches in the presence of a high volume of data (Tsai et al., 2009; Gao et al., 2014; Ashfaq et al., 2017).

As mentioned earlier, deep learning models are vulnerable to attacks. Two major categories of these

<sup>a</sup> <https://orcid.org/0000-0002-0742-2324>

<sup>b</sup> <https://orcid.org/0000-0002-1240-6433>

attacks in the network domain are poisoning and evasion attacks (Peng et al., 2019). In the poisoning attack, the goal is to manipulate the training data and degrade the model performance. In contrast, the evasion attack happens during the test phase, and the attacker aims to fool the DNN model by making small changes to the samples (Pitropakis et al., 2019).

The main contributions of this paper can be summarized as follows:

- This paper evaluates the poisoning attack through label flipping in the network attack classification problem.
- Two well-known, and comprehensive datasets, namely CSE-CIC-IDS2018 and CIC-IDS2017 (Sharafaldin et al., 2018) were used for our detailed experiments.
- To understand the relation between the amount of flipped labels and datasets' characteristics with the attack's effect on the trained model, several experiments in different settings were performed.

The rest of this paper is organized as follows: Section three reviews the related works, and Section four describes the proposed method. Section four presents the experimental settings, and Section six contains a detailed analysis discussion of the results. Section seven concludes the paper.

## 2 RELATED WORKS

There are different adversarial attacks for machine learning phases. Poisoning attacks are applied during the training phase and include manipulating training data (Schwarzschild et al., 2021). One of the poisoning attack methods is data injection. In data injection, the attacker injects malicious inputs into the training set to affect the model's decision boundaries. In a label-flipping attack, the labels of the samples in the training set are changed while the features are remained unchanged (Tabassi et al., 2019).

The attacks during the test phase are called evasion attacks. Evasion attacks aim to craft malicious inputs that can fool trained machine-learning models. These crafted inputs are called adversarial examples. These examples can fool deep learning models into making wrong decisions, while a human observer can correctly classify these examples (Goodfellow et al., 2014; Papernot et al., 2017). Goodfellow et al. proposed a fast and simple method for generating adversarial examples. They called their method Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014). Some other well-known evasion attack methods are JSMA (Papernot et al., 2016), C&W attack (Carlini

and Wagner, 2017), and Deepfool (Moosavi-Dezfooli et al., 2016).

Most research on adversarial attacks in the network domain focuses on evasion attacks. One of the first attempts to perform an evasion attack on network datasets was in (Warzyński and Kołaczek, 2018). They used the FGSM technique to generate adversarial examples on the NSL-KDD dataset during test time. In (Wang, 2018), Wang did comprehensive experiments on the NSL-KDD dataset using FGSM, JSMA, Deepfool, and C&W techniques and also measured the effect of the different selected features on evasion attack success.

To the best of our knowledge, few works focus on adversarial attacks during the training phase for machine learning-based network intrusion systems. In (Apruzzese et al., 2019), Apruzzese et al. made a label-flipping attack against a botnet dataset. In their attack, they selected existing malicious samples and slightly changed the value of three features, including *duration*, *exchanged\_bytes*, and *total\_packets*. Then flipped the label to benign. They performed the attack against Random Forest, Multi-layer Perceptron, and KNN. Their experiments show that the model performance drops significantly after the poisoning attack. Papadopoulos et al. made a label-flipping attack against SVM based model for the Bot-IoT dataset (Papadopoulos et al., 2021). They sorted the samples based on their distance from the SVM hyperplane and flipped the one with a smaller margin to the hyperplane. Their experiments showed that random and targeted attacks severely affect the classification metrics.

One of the main areas for improvement of the previous works is focusing on traditional machine-learning techniques for performing adversarial deep-learning attacks. Also, in most cases, researchers used simple and outdated datasets for experimental analysis and evaluation of their proposed models.

## 3 PROPOSED MODEL

This section explains our model for performing the label-flipping attack against the NIDS. To execute the attack scenario, first, we need to train the target model on the selected datasets and conduct the proposed label-flipping attack.

Figure 1 shows the relation of the evasion and poisoning attack concerning the machine learning pipeline. Evasion attack uses the trained model and original samples from the dataset to generate adversarial examples for fooling the deep learning model. But, the poisoning attack happens before the training phase by changing the training set.

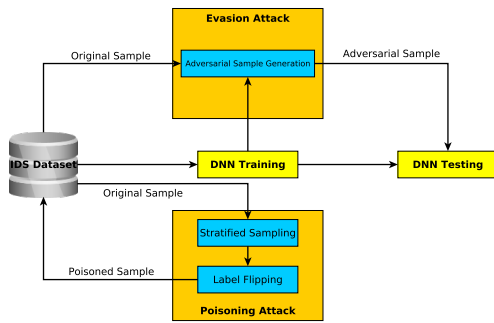


Figure 1: Adversarial Attack in Machine Learning.

### 3.1 Training the DNN Target Model

First, we train our DNN model for classifying different network attacks. We train a multi-layer perceptron with two hidden layers, each of them containing 256 neurons. During the training, the inputs are 76 features of each network flow, and the outputs are different probability values for each attack type and benign flow. We used ReLU as our activation function and a Dropout layer with 0.2 probability in both hidden layers.

### 3.2 Label Flipping Attack

In the label-flipping attack, the attacker's goal is to change the labels of the samples in the training set to decrease the performance of the deep learning model. There are two steps in this attack. First, select the subset of training samples and then flip the labels of selected samples.

In probability sampling, each record in the dataset has the same possibility of being selected. There are four main types of probability sampling (McCombes, 2022).

- **Simple Random Sampling.** In this method, each record has an equal chance of being selected. The whole population is the sampling frame, and each sample is selected with the same probability.
- **Systemic Sampling.** In systemic sampling, samples are selected at regular intervals starting from a random start point. Since we specified the sample from a list with the same intervals, we should be careful that there is no hidden pattern in the provided list which may cause the sample to be skewed.
- **Stratified Sampling.** This method divides the population into subgroups (strata) based on their relevant characteristics. The number of samples that should be selected from each subgroup is calculated based on the population proportion they

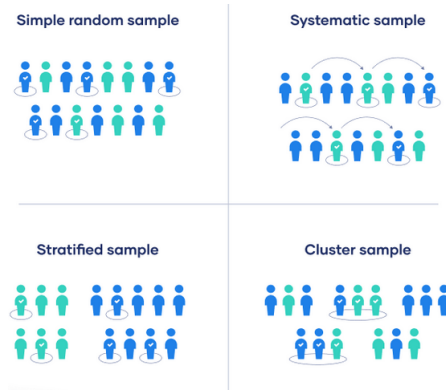


Figure 2: Different Probability Sampling Methods (McCombes, 2022).

include. Then, the samples are chosen using one simple or systemic sampling method.

- **Cluster Sampling.** In cluster sampling, the population is divided into subgroups, but each subset should have the same characteristics as the whole sample. Then, either an entire cluster is selected as the sample, or one of the other sampling methods is used to choose samples from the selected cluster.

Since we want to have samples from all the classes in the label-flipping attack using stratified sampling seems appropriate. We divide the training set samples into subgroups based on their class label and use simple random sampling to select samples from each subgroup. Then, we change the selected sample label to any label other than its actual label.

## 4 EXPERIMENTS

First, we train our DNN model for classifying network attacks in the selected datasets and demonstrate the classifier's performance. Then, the label-flipping attack is performed using the above procedure. To better understand the attack, we start with flipping 10 percent of the labels and go up to 70 percent. In each step, we keep the previously flipped labels and select another 10 percent of actual labels to do the label flipping. We do the attack for different percentages of the flipped labels ten times and report our results' average and standard deviation. Also, from those ten various experiments, the one with the less effect on the model's performance is selected and used in the next step.

In the selected datasets, the ratio of malicious to benign traffic is 20 percent to 80 percent. To evaluate the effect of different ratios on the deep learning model performance in the presence of label flipping

Table 1: Records in CIC-IDS2017 dataset.

Attack Name	Number of Records
Benign	2271320
DDoS	128025
PortScan	158804
Botnet	1956
Infiltration	36
Web Attack-Brute Force	1507
Web Attack-Sql Injection	21
Web Attack-XSS	652
FTP-Patator	7935
SSH-Patator	5897
DoS GoldenEye	10293
DoS Hulk	230124
DoS Slowhttp	5499
Dos Slowloris	5796
Heartbleed	11

Table 2: Records in CSE-CIC-IDS2018 dataset.

Attack Name	Number of Records
Benign	13390249
Bot	286191
Infiltration	160639
Brute Force-Web	611
Brute Force-XSS	230
SQL Injection	87
FTP-BruteForce	193354
SSH-BruteForce	187589
DoS GoldenEye	41508
DoS Hulk	461912
DoS Slowhttp	139890
Dos Slowloris	10990
DDoS LOIC-HTTP	576191
DDoS LOIC-UDP	1730
DDoS HOIC	686012

attack, we adjust the balance between the malicious and benign traffic from 20-80 to 30-70, 40-60, 50-50, 60-40, 70-30, and 80-20 in the preformed experiences.

### 4.1 Dataset

CIC-IDS2017 (Sharafaldin et al., 2018) and CSE-CIC-IDS2018 are used to train the DNN model and perform the label-flipping attack. Each dataset contains several network attacks. CSE-CIC-IDS2018 is an extension to CIC-IDS2017 and contains the same kind of attacks. They extracted more than 80 network traffic features from their datasets using CICFlowMeter (Lashkari et al., 2017) and labeled each Flow as benign or attack name. The details of these two datasets can be found in Table 1 and Table 2.

### 4.2 Target Model Training

We train a DNN model on both datasets and compare its performance with other machine learning techniques. The DNN model is a simple multi-layer perceptron. Table 3 shows the results, which present that our model has comparable performance with other machine learning models.

Table 3: Results of the Classifiers.

Machine Learning Techniques	IDS2017			IDS2018		
	F1-score	PC	RC	F1-score	PC	RC
DT	99.84	99.76	99.92	97.70	99.73	96.14
Naive Bayes	28.47	32.43	73.75	48.29	49.48	72.79
LR	36.71	39.76	34.96	57.97	64.13	56.84
RF	96.58	99.79	94.24	94.23	99.81	90.97
<b>DNN (Our)</b>	97.99	92.98	88.09	97.97	97.51	97.46

the dataset. The other main rows are label-flipping attack results with varying percentages of flipped labels from 10 to 70.

As shown in Table 4, increasing the percentage of flipped labels decreases the performance of the deep learning model. The average accuracy is 98.06% without flipped labels, dropping to 29.25% when 70 percent of the labels are flipped. Figure 3 shows the model’s performance for the dataset with the 20-80 ratio and different percentages of the flipped label. All four metrics drop significantly when the percentage increases from 10 to 70.

With the same percentage of the flipped labels, when we change the ratio of malicious to benign samples, the accuracy stays almost the same, but the recall and F1-score increase. In the beginning, when the distribution is 20-80, most of the selected labels for flipping attacks are from the benign class, and the number of actual benign samples is high enough for the model to learn and detect them. But, with changing the training dataset toward increasing the ratio of malicious samples, the model’s ability to detect the benign samples in the presence of label flipping attack drops while its performance for other classes increases, and since in the multi-class classification, the average is used to calculate the recall, this metric will increase. Figure 4 shows the results for the attack with 40% flipped labels with different malicious to benign ratios. As it is clear, with changing the distribution, accuracy stays almost without change, but recall and F1-score increase.

The same experiments have been done for the

## 5 ANALYSIS AND DISCUSSION

In this Section, we will analyze the experimental results and compare the findings for the selected datasets with different percentages of the flipped label and malicious to benign ratio.

Table 4 presents the results for the CIC-IDS2017 dataset. The first main row is the classifier’s performance without label flipping, and each sub-row is for the different ratio of the malicious to benign traffic in

Table 4: CIC-IDS2017 Results.

Selected Labels	Ratio	Accuracy	Precision	Recall	F1-score
0	20-80	97.66	95.47	87.82	89.84
	30-70	98.05	95.83	87.52	89.99
	40-60	97.94	93.97	91.06	91.72
	50-50	98.14	95.22	94.59	94.82
	60-40	97.93	96.38	91.99	93.68
	70-30	98.17	97.03	90.28	92.82
	80-20	98.55	95.62	94.64	94.84
	20-80	88.08±0.21	77.62±1.57	38.44±0.37	43.28±0.38
10	30-70	87.94±0.20	79.28±2.53	43.46±0.12	48.60±0.15
	40-60	87.84±0.16	78.90±2.58	46.80±0.16	52.25±0.26
	50-50	87.87±0.08	79.37±0.52	49.29±0.12	54.91±0.19
	60-40	87.99±0.02	78.76±0.48	51.69±0.12	57.50±0.19
	70-30	88.26±0.03	81.68±4.19	53.52±0.06	59.41±0.07
	80-20	88.55±0.06	79.50±2.27	55.72±0.81	61.53±0.72
	20-80	78.03±0.06	66.42±3.32	30.86±0.27	33.62±0.17
	30-70	77.96±0.17	69.91±0.71	35.49±0.20	38.07±0.20
20	40-60	77.86±0.06	70.35±2.06	38.48±0.08	41.23±0.12
	50-50	77.98±0.06	70.02±0.80	40.68±0.12	43.62±0.21
	60-40	78.15±0.02	70.20±0.64	42.55±0.12	45.73±0.20
	70-30	78.43±0.03	70.43±0.52	44.17±0.06	47.63±0.10
	80-20	78.64±0.04	70.65±0.20	45.38±0.04	49.07±0.04
	20-80	68.23±0.7	58.05±2.74	26.33±0.14	27.75±0.13
	30-70	68.09±0.04	60.57±2.93	30.64±0.08	31.34±0.09
	40-60	68.07±0.06	61.39±1.51	33.41±0.04	33.99±0.12
30	50-50	68.22±0.02	61.59±0.66	35.53±0.9	36.15±0.17
	60-40	68.36±0.02	61.08±0.44	37.25±0.04	38.06±0.07
	70-30	68.59±0.02	61.80±0.44	38.54±0.06	39.50±0.11
	80-20	68.80±0.02	61.68±0.16	39.51±0.02	40.76±0.03
	20-80	58.48±0.07	47.42±1.67	22.77±0.08	23.05±0.03
	30-70	58.29±0.08	52.06±2.63	26.67±0.14	26.10±0.07
	40-60	58.34±0.05	50.18±2.47	29.43±0.05	28.38±0.09
	50-50	58.40±0.03	52.86±1.49	31.37±0.04	30.07±0.07
40	60-40	58.57±0.04	51.92±1.44	32.97±0.08	31.63±0.15
	70-30	58.77±0.02	53.22±0.48	34.17±0.04	32.98±0.06
	80-20	58.94±0.04	52.78±0.30	34.89±0.05	33.98±0.08
	20-80	48.69±0.03	39.01±0.1	19.69±0.11	18.84±0.06
	30-70	48.54±0.07	41.13±2.71	23.18±0.14	21.51±0.05
	40-60	48.58±0.04	40.08±1.58	25.84±0.03	23.43±0.05
	50-50	48.65±0.02	43.70±2.17	27.66±0.02	24.83±0.03
	60-40	48.76±0.02	42.24±1.62	29.12±0.05	26.09±0.11
50	70-30	48.96±0.02	53.22±0.48	34.17±0.04	32.98±0.06
	80-20	49.07±0.04	43.98±0.18	30.68±0.04	27.96±0.05
	20-80	38.96±0.02	31.19±0.23	17.00±0.05	14.96±0.05
	30-70	38.84±0.04	32.04±1.45	19.93±0.13	17.19±0.07
	40-60	38.90±0.04	31.64±0.12	22.30±0.02	18.82±0.05
	50-50	38.95±0.02	33.49±2.02	23.98±0.01	20.00±0.03
	60-40	39.03±0.01	32.83±1.51	25.27±0.01	20.96±0.02
	70-30	39.13±0.03	33.54±1.54	26.13±0.05	21.79±0.10
60	80-20	39.20±0.03	34.97±0.16	26.48±0.02	22.37±0.04
	20-80	29.22±0.06	22.91±1.22	14.41±0.1	11.13±0.09
	30-70	29.15±0.03	24.40±2.94	16.65±0.05	12.94±0.11
	40-60	29.17±0.02	23.80±0.11	18.84±0.06	14.19±0.07
	50-50	29.25±0.01	25.03±1.84	20.00±0.01	15.21±0.02
	60-40	29.27±0.02	25.55±2.86	21.03±0.01	15.93±0.03
	70-30	29.35±0.01	25.19±1.37	21.69±0.01	16.52±0.02
	80-20	29.40±0.02	27.02±2.93	21.92±0.02	16.95±0.06

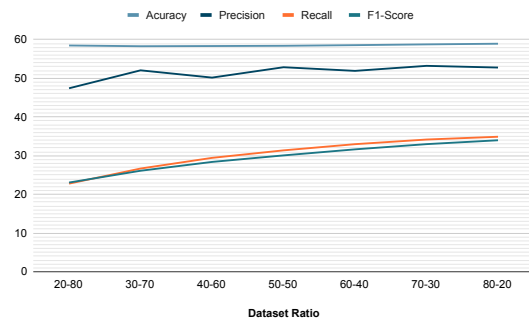


Figure 4: CIC-IDS2017 Results with 40% Flipped Labels.

creased. The average accuracy drops from 97.36% with no flipped labels to 29.2% when 70% of the labels are flipped. In Figure 5, decreasing performance metrics are clearly shown for the dataset with the 20-80 ratio.

The results for each percentage of the flipped label are the same as CIC-IDS2017. The accuracy stays almost unchanged, and the F1-score and recall increase significantly. In Figure 6, you see how accuracy, precision, recall, and F1-score change when the ratio of the dataset changes from 20-80 to 80-20.

Analysis and comparison of the results for both datasets show that, as expected, the label-flipping attack severely affects the deep learning model. Increasing the number of flipped labels decreases the model's performance significantly. Also, the accuracy stays the same with changing the ratio of the samples in the dataset. But, some improvements happen in recall and F1-score.

## 6 CONCLUSION

With the increase of using deep learning models in network security problems, these models' vulnerabilities have gained more attraction among researchers. Deep learning models are susceptible to several attacks, including evasion attacks and poisoning attacks. Many works tried to evaluate evasion attacks in the network security domain, but there needs to be studied regarding poisoning attacks.

This paper investigates the effect of label-flipping attacks in the network domain. CSE-CIC-IDS2018 and CIC-IDS2017 (Sharafaldin et al., 2018), two well-known network intrusion detection datasets, are selected for performing the label-flipping attack. After the preprocessing step, we trained a DNN model as the target model for the attack. We select a subset of samples in each step using stratified sampling to flip the labels. The attack was made by changing the percentage of the flipped labels from 10 to 70 for the

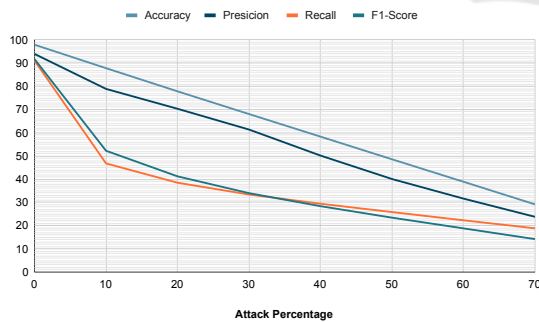


Figure 3: CIC-IDS2017 Results for Dataset with 20-80 Malicious to Benign Ratio.

CSE-CIC-IDS2018 dataset, and the results are reported in Table 5. The percentage of flipped labels are increased from 10 to 70, and for each value, the experiments are done with different ratio of the malicious to benign samples. Similar to the results of the CIC-IDS2017 dataset, the model's performance drops when the percentage of the flipped labels is in-

Table 5: CSE-CIC-IDS2018 Results.

Selected Labels	Ratio	Accuracy	Precision	Recall	F1-score
0	20-80	95.87	91.53	81.64	83.58
	30-70	98.27	91.80	88.69	89.48
	40-60	97.96	91.13	92.03	90.97
	50-50	97.66	91.71	92.17	91.51
	60-40	97.71	92.46	94.02	92.82
	70-30	97.05	92.19	92.41	91.81
	80-20	97.01	92.54	94.07	92.94
	20-80	88.71±0.27	82.66±0.74	46.49±0.62	53.16±0.44
10	30-70	88.32±0.15	82.75±0.52	55.61±0.34	60.11±0.31
	40-60	87.90±0.16	82.62±0.51	59.22±0.33	62.81±0.38
	50-50	87.48±0.21	82.79±0.58	61.96±0.16	64.95±0.29
	60-40	87.91±0.04	83.17±0.25	65.58±0.09	68.60±0.12
	70-30	87.26±0.08	83.09±0.41	65.94±0.27	68.57±0.38
	80-20	87.29±0.05	83.65±0.36	68.27±0.10	70.69±0.13
	20-80	78.87±0.19	73.21±0.42	35.69±0.49	41.71±0.36
	30-70	78.40±0.11	73.41±0.39	44.97±0.12	49.20±0.15
20	40-60	78.15±0.03	73.31±0.31	49.42±0.08	52.50±0.11
	50-50	77.86±0.08	73.56±0.37	52.51±0.18	54.90±0.18
	60-40	78.05±0.4	73.91±0.17	55.87±0.08	57.90±0.13
	70-30	77.47±0.07	73.66±0.37	57.09±0.10	58.46±0.15
	80-20	77.56±0.05	74.34±0.38	59.30±0.10	60.55±0.14
	20-80	68.99±0.16	63.49±0.90	28.79±0.27	33.39±0.34
	30-70	68.63±0.06	64.59±0.32	37.44±0.02	40.83±0.06
	40-60	68.36±0.11	64.40±0.39	41.83±0.13	44.15±0.18
30	50-50	68.03±0.14	64.38±0.19	45.00±0.16	46.49±0.18
	60-40	68.27±0.02	64.58±0.21	48.41±0.04	49.38±0.07
	70-30	67.76±0.09	64.48±0.15	49.93±0.12	50.14±0.13
	80-20	67.79±0.08	64.77±0.16	51.96±0.10	51.99±0.16
	20-80	58.98±0.44	54.21±1.12	23.40±0.71	26.45±0.88
	30-70	58.81±0.02	55.24±0.26	31.35±0.04	33.64±0.06
	40-60	58.53±0.09	55.21±0.29	35.38±0.11	36.75±0.11
	50-50	58.31±0.03	55.18±0.18	38.59±0.05	39.10±0.09
40	60-40	58.49±0.07	55.64±0.33	41.79±0.08	41.70±0.07
	70-30	58.04±0.10	55.28±0.05	43.44±0.15	42.61±0.16
	80-20	58.11±0.03	55.47±0.03	45.30±0.04	44.27±0.08
	20-80	48.88±0.35	44.62±0.89	19.27±0.49	20.51±0.69
	30-70	48.99±0.6	45.95±0.22	26.20±0.06	27.20±0.10
	40-60	48.72±0.11	45.86±0.33	29.73±0.12	29.99±0.12
	50-50	48.58±0.05	46.09±0.18	32.71±0.05	32.16±0.08
	60-40	48.70±0.06	46.49±0.28	35.56±0.07	34.48±0.05
50	70-30	48.33±0.09	46.23±0.11	37.14±0.11	35.35±0.11
	80-20	48.45±0.01	46.36±0.04	38.83±0.01	36.87±0.01
	20-80	39.23±0.35	36.02±0.85	16.40±0.47	16.05±0.61
	30-70	39.05±0.33	36.67±0.49	21.52±0.36	21.09±0.48
	40-60	39.00±0.07	36.85±0.20	24.65±0.07	23.74±0.08
	50-50	38.87±0.07	36.85±0.18	27.16±0.07	25.61±0.06
	60-40	38.99±0.01	37.22±0.12	29.55±0.01	27.55±0.02
	70-30	38.72±0.06	37.21±0.23	30.98±0.07	28.41±0.05
60	80-20	38.81±0.02	36.95±0.85	32.30±0.02	29.64±0.05
	20-80	29.40±0.12	26.46±1.08	13.82±0.15	11.63±0.25
	30-70	29.18±0.28	27.13±0.66	17.37±0.29	15.47±0.36
	40-60	29.15±0.29	27.70±0.35	19.68±0.28	17.60±0.29
	50-50	29.15±0.08	28.00±0.17	21.66±0.07	19.16±0.12
	60-40	29.30±0.01	27.95±0.72	23.49±0.01	20.75±0.02
	70-30	29.04±0.04	28.00±0.19	24.49±0.04	21.47±0.05
	80-20	29.18±0.00	28.11±0.02	25.48±0.00	22.48±0.01

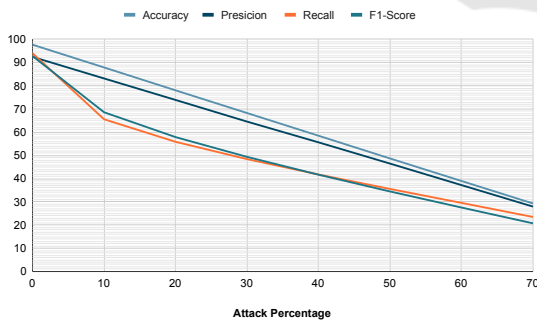


Figure 5: CSE-CIC-IDS2018 Results for Dataset with 20-80 Malicious to Benign Ratio.

different ratios of the malicious to benign samples in the datasets.

The reported results show that, as expected, the label flipping attack can severely affect the deep learning-based IDS performance, and some measures should be in place to defend against these attacks. Also, the malicious to the benign ratio of the samples

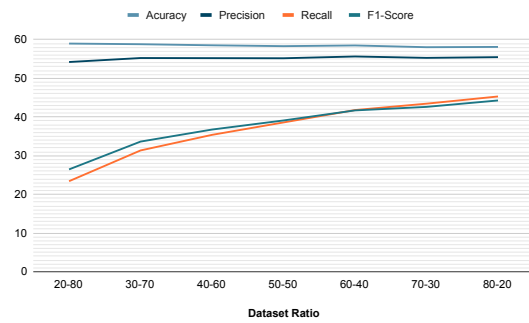


Figure 6: CSE-CIC-IDS2018 Results with 40% Flipped Labels.

in the datasets may have minor effects on the attack’s success. Still, in general, the model’s performance always drops in the presence of the label-flipping attack.

While the researchers are showing more interest in using deep learning models in network intrusion detection systems, they should pay more attention to the security of these models and their vulnerabilities against evasion and poisoning attacks. In (Mohammadian et al., 2022; Mohammadian et al., 2023), we evaluated the deep learning-based network intrusion detection systems in the presence of evasion attacks, and here we did the same for the label-flipping attack. For future work, we will combine these attacks into a more complex, sophisticated framework for attacking deep learning-based NIDS. Also, researchers should work on finding techniques to defend against these attacks and help maintain the model’s performance.

## REFERENCES

Apruzzese, G., Colajanni, M., Ferretti, L., and Marchetti, M. (2019). Addressing adversarial attacks against security systems based on machine learning. In *2019 11th international conference on cyber conflict (Cy-Con)*, volume 900, pages 1–18. IEEE.

Ashfaq, R. A. R., Wang, X.-Z., Huang, J. Z., Abbas, H., and He, Y.-L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 378:484–497.

Biggio, B. and Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331.

Buczak, A. L. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. (2021). Label-only membership inference at-

- tacks. In *International conference on machine learning*, pages 1964–1974. PMLR.
- Dalvi, N., Domingos, P., Sanghai, S., and Verma, D. (2004). Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108.
- Duddu, V. (2018). A survey of adversarial machine learning in cyber warfare. *Defence Science Journal*, 68(4).
- Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333.
- Gao, N., Gao, L., Gao, Q., and Wang, H. (2014). An intrusion detection model based on deep belief networks. In *2014 Second International Conference on Advanced Cloud and Big Data*, pages 247–252. IEEE.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., and Zhang, X. (2022). Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37.
- Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., and Ghorbani, A. A. (2017). Characterization of tor traffic using time based features. In *ICISSP*, pages 253–262.
- McCombes, S. (2022). Sampling methods — types, techniques & examples. <https://www.scribbr.com/methodology/sampling-methods/>.
- Mohammadian, H., Ghorbani, A. A., and Lashkari, A. H. (2023). A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems. *Applied Soft Computing*, 137:110173.
- Mohammadian, H., Lashkari, A. H., and Ghorbani, A. A. (2022). Evaluating deep learning-based nids in adversarial settings. In *ICISSP*, pages 435–444.
- Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.
- Papadopoulos, P., Thornewill von Essen, O., Pitropakis, N., Chrysoulas, C., Mylonas, A., and Buchanan, W. J. (2021). Launching adversarial attacks against network intrusion detection systems for iot. *Journal of Cybersecurity and Privacy*, 1(2):252–273.
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519.
- Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE.
- Peng, Y., Su, J., Shi, X., and Zhao, B. (2019). Evaluating deep learning based network intrusion detection system in adversarial environment. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 61–66. IEEE.
- Pitropakis, N., Panaousis, E., Giannetsos, T., Anastasiadis, E., and Loukas, G. (2019). A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 34:100199.
- Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., and Goldstein, T. (2021). Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR.
- Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116.
- Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Tabassi, E., Burns, K. J., Hadjimichael, M., Molina-Markham, A. D., and Sexton, J. T. (2019). A taxonomy and terminology of adversarial machine learning. *NIST IR*, pages 1–29.
- Tian, Z., Cui, L., Liang, J., and Yu, S. (2022). A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35.
- Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. (2018). Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*.
- Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., and Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. *expert systems with applications*, 36(10):11994–12000.
- Wang, Z. (2018). Deep learning-based intrusion detection with adversaries. *IEEE Access*, 6:38367–38384.
- Wang, Z., Ma, J., Wang, X., Hu, J., Qin, Z., and Ren, K. (2022). Threats to training: A survey of poisoning attacks and defenses on machine learning systems. *ACM Computing Surveys*, 55(7):1–36.
- Warzyński, A. and Kołaczek, G. (2018). Intrusion detection systems vulnerability on adversarial examples. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–4. IEEE.
- Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261.