# Low-Quality Training Data Only?

# A Robust Framework for Detecting Encrypted Malicious Network Traffic

Yuqi Qing, Qilei Yin, Xinhao Deng, Yihao Chen, Zhuotao Liu, Kun Sun, Ke Xu, Jia Zhang, Qi Li

NDSS SYMPOSIUM/2024

Presented by Internet Society

#NDSSSymposium2024

# ML/DL-based Encrypted Malicious Traffic Detection

- ML/DL is effective in detecting encrypted malicious traffic
  - Traditional detection methods focus on analyzing plaintext payloads, which is useless when facing encrypted traffic.
  - ML/DL models can capture the essential characteristics of encrypted malicious.
- Most ML/DL-based methods rely on high-quality training datasets.
- However, collecting high-quality data is time-consuming and difficult.

What if we only have **Low-quality** training data?

# What's Low-Quality Training Data

- Having Non-negligible Label Noises
  - Public Service's (e.g., Virustotal) labels are different from year to year, which are not always reliable.
  - Manually Labeling incurs large overheads, especially when labeling encrypted traffic.

- Insufficient Malicious Training Samples
  - The typical approach for collecting malicious training samples is to execute malware samples captured in real-world cyberspace in controlled sandboxes and collect the generated traffic.
  - But malware in the real world keeps evolving. Collecting time-sensitive malware data is difficult.

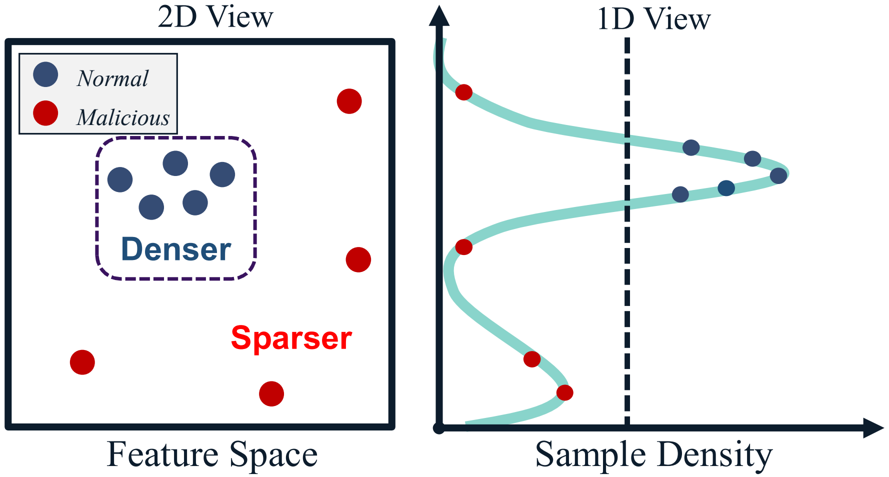
**Low-quality training data can degrade the detection performance of ML/DL-based encrypted malicious traffic detection methods.**

# Potential Solution?

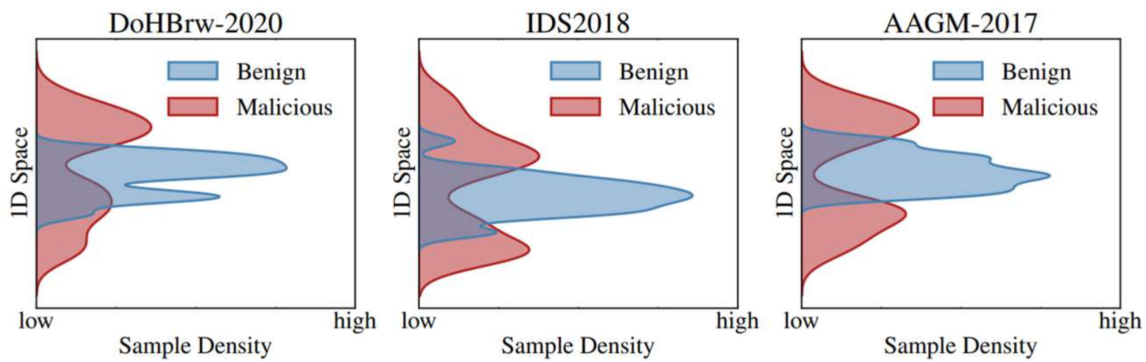
- Robust Machine Learning Models
  - Rely on strong assumptions or prior knowledge.
- Data Augmentation Methods
  - Label noises will confuse the distributions of different categories, resulting in synthesizing more label noises.
- Pre-training an DL model using large-scale unlabeled encrypted training data
  - Collecting and pre-processing a large-scale dataset is expensive. It may also increase the risk of privacy leakage.
- **Our Goal:** **Detecting encrypted malicious traffic accurately using only a low-quality training set.**

# Key Observation

The distribution of benign data is **denser**, while the distribution of malicious data is **sparser**.

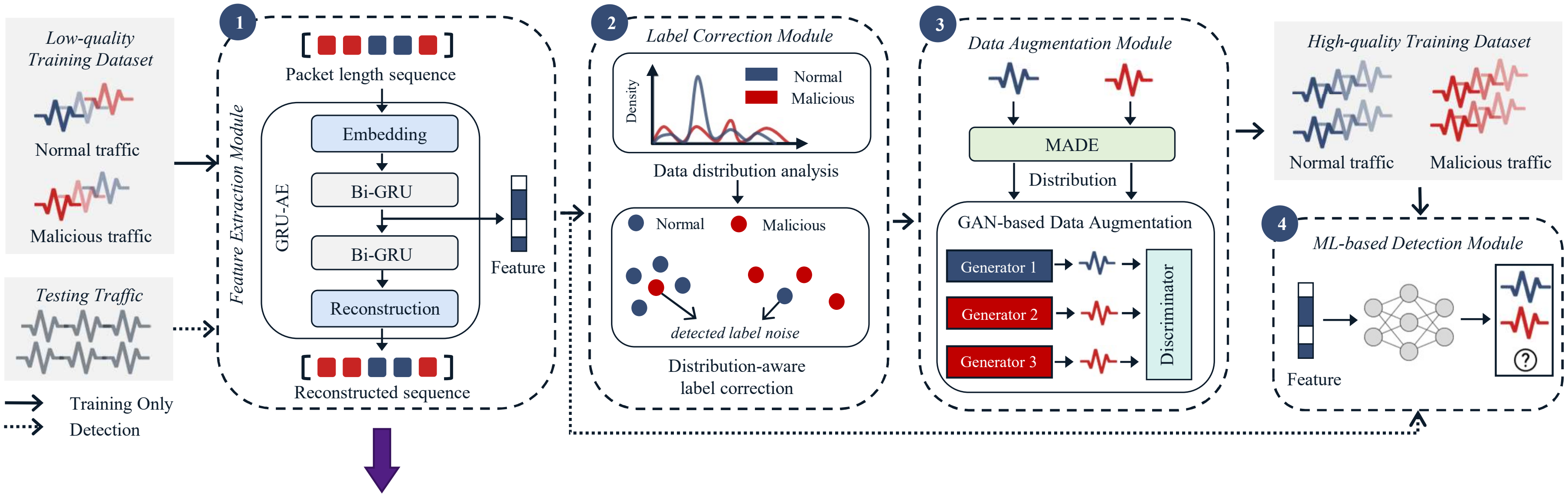



Pre-experiments: Data distribution on 3 datasets
(utilizing t-SNE to reduce dimension)



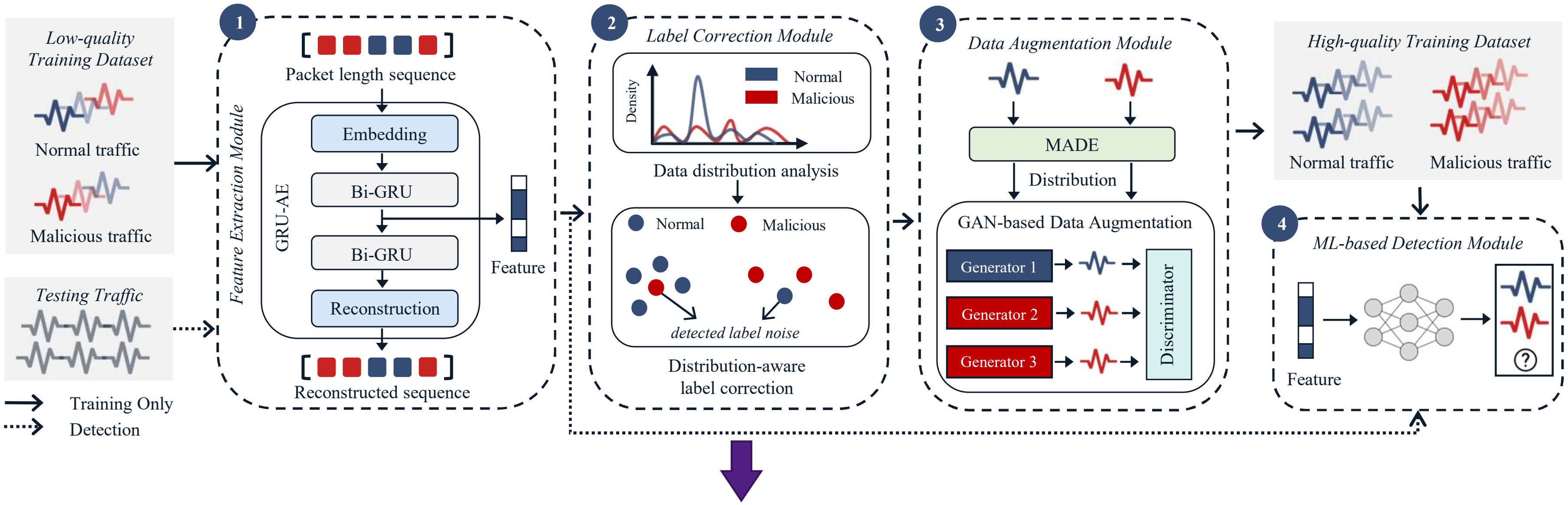

- **Core Idea:** **Leveraging the difference in distribution between benign and malicious traffic data** to infer the true label of training samples and generate new training data that can represent new encrypted malicious traffic

# Framework Overview



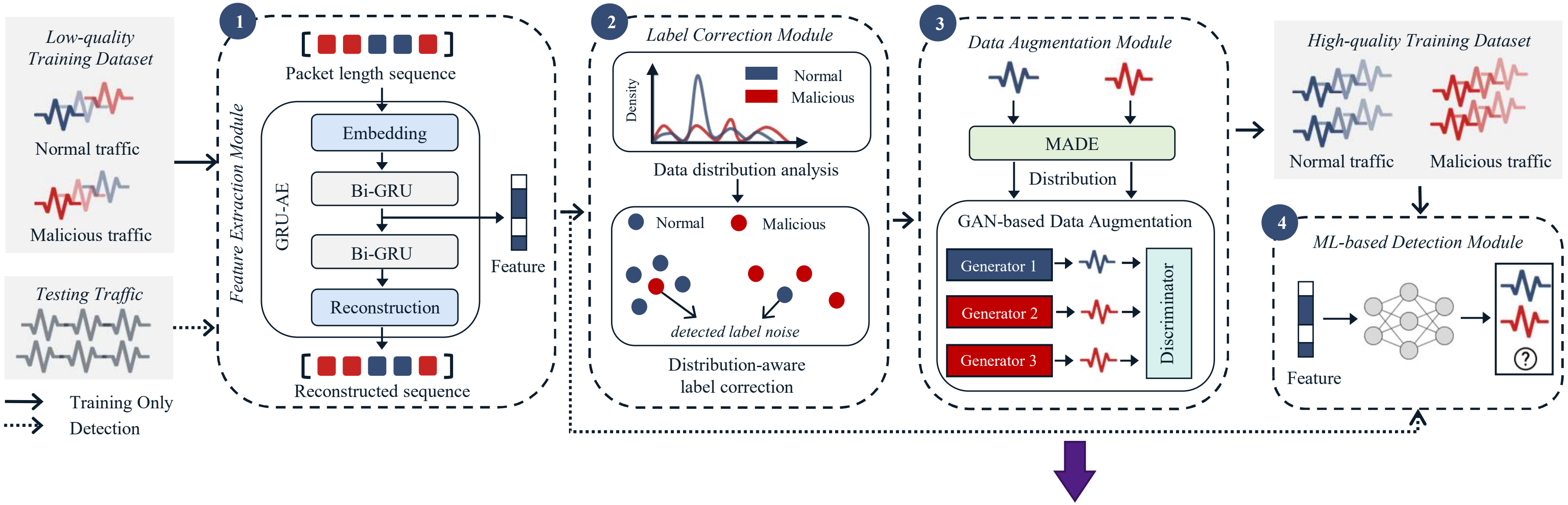

**Feature Extraction Module:** Convert the raw encrypted network traffic into feature vectors

# Framework Overview




**Label Correction Module:** Estimate training data's distributions to detect and correct label noises

# Framework Overview

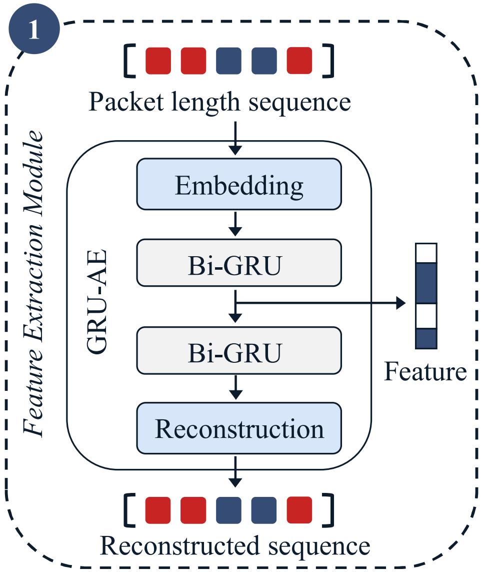



**Data Augmentation Module:** Generate new training data that can represent new encrypted malicious traffic

# Feature Extraction Module

**Goal:** Convert the raw encrypted network traffic into feature vectors

**Challenge:** Needs to handle traffic encrypted by various types and versions of encryption protocols & The non-negligible label noises will result in inaccurate feature selection.



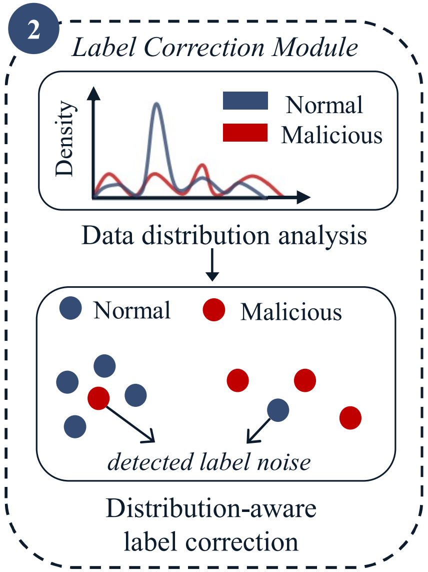

Using an **Sequential Auto-Encoder** model to automatically learn the most representative features of input data in an unsupervised manner and minimize the effects of label noises.

- Using Packet length sequence (of a traffic flow) as input data
- Using Gated Recurrent Units (GRU) to learn the sequential features of traffic from packet length sequences.

# Label Correction Module

**Goal:** Estimating training data's distributions to detect and correct label noises

**Challenge:** High-dimensional traffic data's distribution is hard to accurately estimate & The difference between normal and malicious data distributions may not be significant




Using a **Deep Generative Model** to accurately estimate the data's distribution, relabeling part of the training samples that have the most significant distributions, and utilizing them to infer other samples' labels through ensemble learning

- Using MADE[1] model to estimate traffic data's distribution.
- Using probability density to identify the training samples that have the most significant distributions.
- Building an ensemble of seven classical machine learning classifiers.

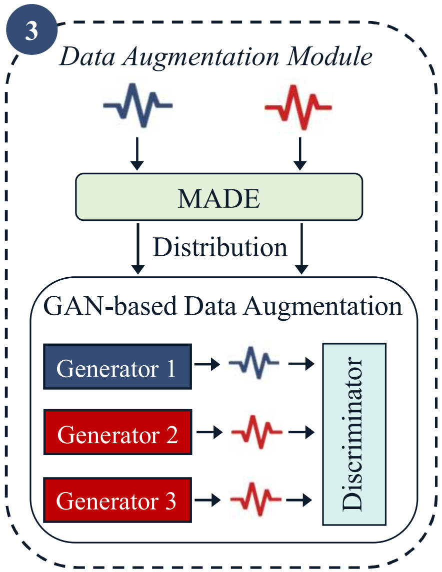


[1] Germain M, Gregor K, Murray I, et al. Made: Masked autoencoder for distribution estimation[C] International conference on machine learning. PMLR, 2015: 881-889.

# Data Augmentation Module

**Goal:** Generate new training data that can represent new encrypted malicious traffic

Challenge The generated data should be diverse, otherwise it may limit the model's generalizability & New malicious samples' distribution may be inconsistent with existing training data

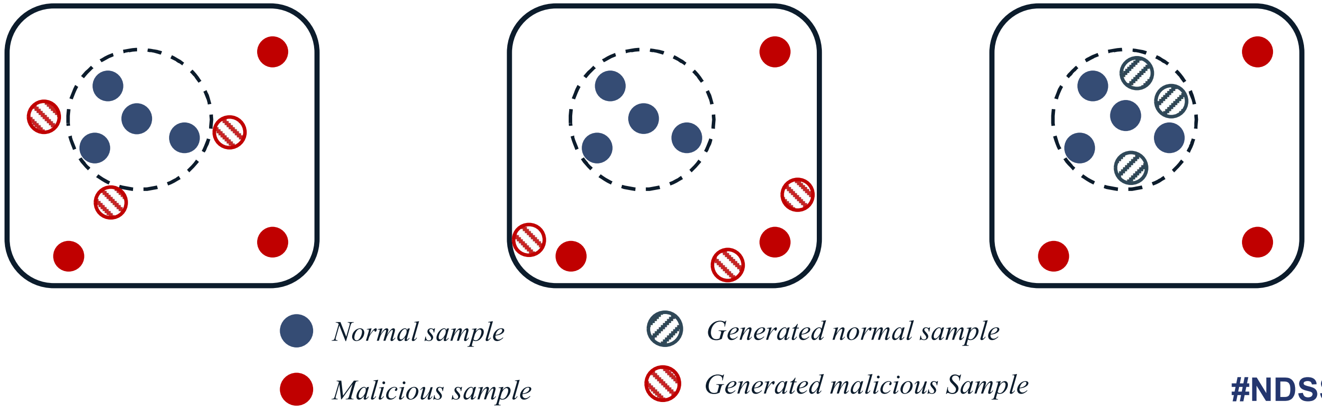



Predicting the potential regions of new malicious samples,
sampling from these target regions to generate new diverse training data

$M_B$: Attackers may mimic normal behaviors to evade detection (malicious similar to normal)

$M_O$: New attack methods are emerging, e.g., Zero-day (malicious that are unique)

$N_B$: To avoid data imbalance, normal samples should also be generated.



# Data Augmentation Module

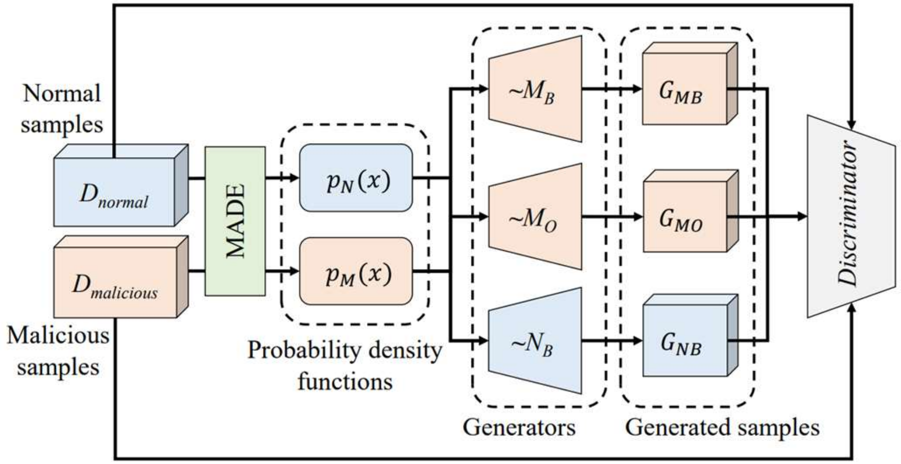
**Goal:** Generate new training data that can represent new encrypted malicious traffic

- Existing augmentation methods can only generate samples similar to existing samples.
- Propose an improved GAN, which can fit 3 specific regions' distribution to generate samples.
  - **Formulate 3 specific regions** with MADE

$$\begin{aligned} M_B &= \{x | p_M(x) < \gamma, \quad \omega_1 \leq p_N(x) < \omega_2\}, \\ M_O &= \{x | p_M(x) < \gamma, \quad 0 \leq p_N(x) < \omega_1\}, \\ N_B &= \{x | p_M(x) < \gamma, \quad \omega_2 \leq p_N(x) < \omega_3\}, \end{aligned}$$

  - Design **3 generators** for 3 regions
  - **New loss function** for each generator

$$\begin{aligned} \mathcal{L}_{KL(X_G||P)} = &- \mathcal{H}(X_G) \\ &+ \mathbb{E}_{x \in X_G, p_M(x) \geq \gamma} \left[\log p_M(x)\right] \\ &- \mathbb{E}_{x \in X_G, p_M(x) < \gamma, p_N(x) < \theta_1} \left[\log p_N(x)\right] \\ &+ \mathbb{E}_{x \in X_G, p_M(x) < \gamma, p_N(x) \geq \theta_2} \left[\log p_N(x)\right], \end{aligned}$$



# Evaluation: Setup

- Datasets: CIRA-CIC-DoHBrw-2020 (DoHBrw)
  CSE-CIC-IDS2018 (IDS)
  IDS (for training)/DoHBrw (for testing)
- Baselines:
  - Traffic Detection: **FS-Net (FS)**[2], **ETA**[3]
  - Robust Detection: **Differential Training (DT)**[4], **ODDS**[5]
  - Robust ML/DL: **SMOTE**[6], **Co-teaching (Co)**[7]

| FS | Co+FS | DT+FS | ODDS+FS | SMOTE+FS | DT+ODDS+FS | ETA | DT+ODDS+ETA |
|---|---|---|---|---|---|---|---|

- Metrics: **F1 score** when benign/malicious ratio is 10:1
  - Such ratio make F1 score results more convincing[8]
- Training label noise ratio: 20%, 25%, 30%, 35%, 40%, 45%
  - Randomly, symmetrically mislabel the samples.
- Training size: each type of samples has 250, 500, or 1000 samples.


[2] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in IEEE INFOCOM, 2019, pp. 1171–1179.

[3] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity," in ACM SIGKDD, 2017, pp. 1723–1732.

[4] J. Xu, Y. Li, and H. D. Robert, "Differential training: A generic framework to reduce label noises for android malware detection," in NDSS, 2021.

[5] S. T. Jan, Q. Hao, T. Hu, J. Pu, S. Oswal, G. Wang, and B. Viswanath, "Throwing darts in the dark? detecting bots with limited data using neural data augmentation," in IEEE S&P, 2020, pp. 1190–1206.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," JAIR, vol. 16, pp. 321– 357, 2002.

[7] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," NeurIPS, vol. 31, 2018.

[8] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "Tesseract: Eliminating experimental bias in malware classification across space and time," in USENIX Security, 2019, pp. 729–746.

# Overall Detection Performance

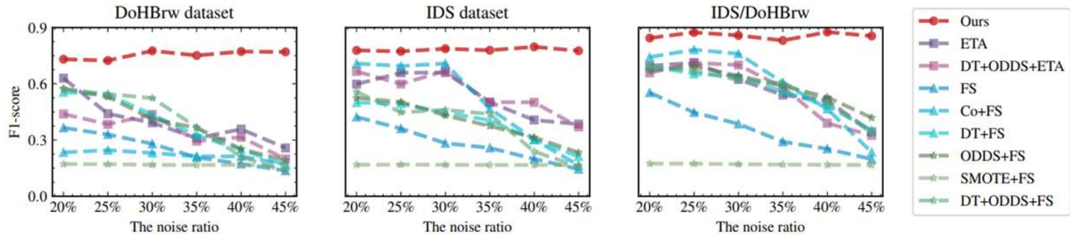



Fig. 6: The F1 score of all methods under different noise ratios on three evaluation datasets.

TABLE III: The F1 score of all methods under different training sizes with the noise ratio of 30%. The F1 score is shown in the form of "Avg ± Std", where Avg is the average F1 score and Std is the standard deviation.

| Method | | FS | Co+FS | DT+FS | ODDS+FS | SMOTE+FS | DT+ODDS+FS | ETA | DT+ODDS+ETA | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| Training Size on DoHBrw | 250 | .18 ± .02 | .21 ± .06 | .41 ± .02 | .42 ± .05 | .17 ± .00 | .34 ± .10 | .56 ± .27 | .44 ± .21 | **.71** ± .02 |
| | 500 | .28 ± .04 | .23 ± .06 | .44 ± .03 | .42 ± .07 | .17 ± .00 | .52 ± .04 | .39 ± .24 | .44 ± .23 | **.78** ± .02 |
| | 1000 | .30 ± .03 | .27 ± .05 | .54 ± .02 | .57 ± .06 | .17 ± .00 | .58 ± .05 | .40 ± .23 | .29 ± .07 | **.78** ± .02 |
| Training Size on IDS | 250 | .26 ± .05 | .66 ± .19 | .34 ± .01 | .46 ± .04 | .17 ± .00 | .38 ± .03 | .57 ± .10 | .53 ± .15 | **.75** ± .04 |
| | 500 | .28 ± .04 | .71 ± .06 | .45 ± .04 | .43 ± .06 | .17 ± .00 | .46 ± .08 | .66 ± .27 | .67 ± .24 | **.79** ± .02 |
| | 1000 | .35 ± .01 | .73 ± .08 | .51 ± .03 | .51 ± .08 | .17 ± .00 | .54 ± .05 | .66 ± .27 | .69 ± .23 | **.77** ± .02 |
| Training Size on IDS/DoHBrw | 250 | .31 ± .13 | .66 ± .19 | .51 ± .10 | .58 ± .12 | .17 ± .01 | .53 ± .11 | .59 ± .12 | .57 ± .15 | **.82** ± .09 |
| | 500 | .39 ± .14 | .76 ± .06 | .64 ± .13 | .64 ± .15 | .17 ± .00 | .62 ± .14 | .62 ± .27 | .70 ± .27 | **.86** ± .05 |
| | 1000 | .38 ± .13 | .73 ± .08 | .68 ± .09 | .69 ± .13 | .17 ± .00 | .72 ± .13 | .69 ± .24 | .73 ± .19 | **.81** ± .07 |

When the noise ratio is 45%, our system still get **F1 scores of 0.770, 0.776, and 0.855,** showing **average improvements of 352.6%, 284.3%, and 214.9%** over the baselines.

# Imitating Real-world Label Noise

- Benign data being mislabeled: **unseen/uncommon domain name**
  - A benign traffic is mislabeled if its domain name is not on Alexa-Top-1m list.
- Malicious data being mislabeled: **absent threat intelligence of a specific type**
  - Choose a malicious type in the dataset, and all samples of the type are mislabeled.

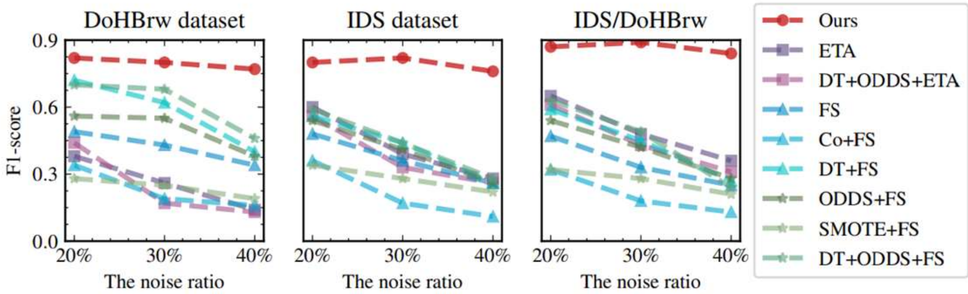



Fig. 7: The F1 scores under realistic label noise settings.

Our system achieves **average F1 of 0.797, 0.800, and 0.867**, achieving **average improvements of 166.5%, 154.6%, and 165.2%** over all baselines.

# Evaluating Label Correction & Data Augmentation Modules

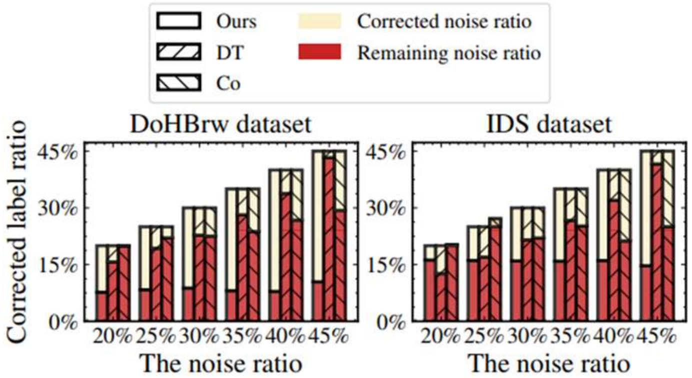



Fig. 8: The corrected and remaining noise ratios under different original noise ratios.

TABLE V: The detection performance after data augmentation. P, R, and F represent precision, recall, and F1 score.

| Dataset | Metrics | NONE | ODDS | SMOTE | GAN | Ours |
|---|---|---|---|---|---|---|
| DoHBrw | P | .88 ± .11 | .86 ± .03 | .82 ± .10 | .85 ± .09 | **.88** ± .03 |
| | R | .93 ± .06 | .93 ± .02 | .95 ± .02 | .93 ± .04 | **.98** ± .01 |
| | F | .90 ± .07 | .89 ± .01 | .87 ± .06 | .88 ± .07 | **.93** ± .01 |
| IDS | P | .67 ± .15 | .44 ± .04 | .46 ± .06 | .48 ± .04 | **.69** ± .12 |
| | R | .86 ± .04 | .89 ± .03 | .89 ± .03 | .87 ± .04 | **.89** ± .02 |
| | F | .77 ± .06 | .74 ± .09 | .59 ± .03 | .60 ± .05 | **.77** ± .06 |
| IDS/DoHBrw | P | .54 ± .14 | .34 ± .10 | .37 ± .08 | .36 ± .06 | **.56** ± .02 |
| | R | .75 ± .09 | .93 ± .02 | .92 ± .00 | .87 ± .04 | **.94** ± .01 |
| | F | .55 ± .09 | .49 ± .11 | .53 ± .07 | .51 ± .05 | **.70** ± .02 |

**Two modules are both more effective than baselines**

# Parameters Sensitivity Analysis

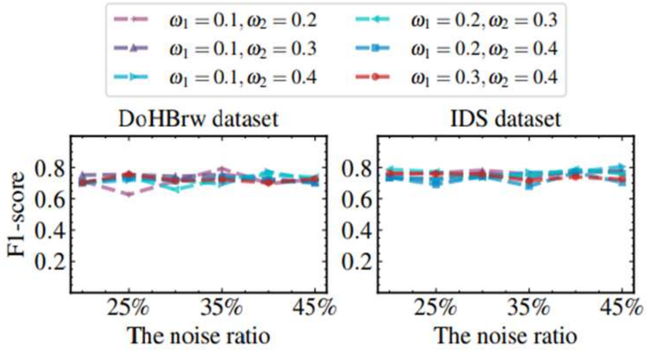



Fig. 10: The F1 score of different thresholds $\omega_1, \omega_2$.

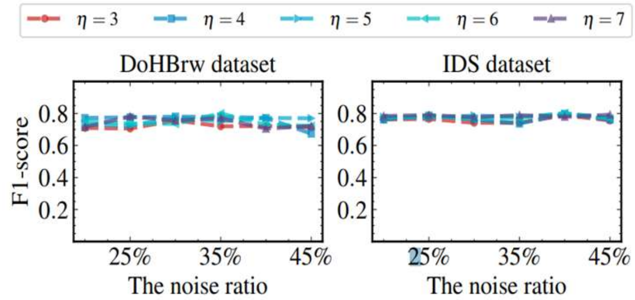



Fig. 11: The F1 score of different number of GAN models $\eta$.

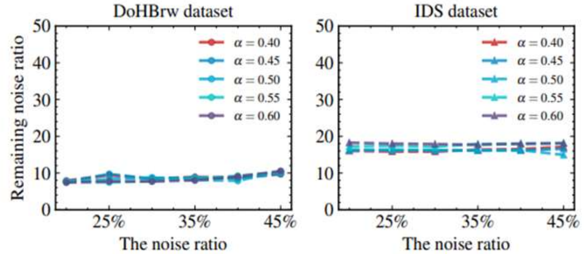



Fig. 9: The remaining noise ratios of various filtering ratios $\alpha$.

control the size and location of
target regions for generated samples

minimize the model collapse of GAN

In label correction module

**Our framework is insensitive to all hyper-parameters**

# Real-world Experiments

- In total, we obtain over 2,900,000 benign and 790,000 malicious encrypted traffic flows with timestamps ranging from Nov. 2017 to Feb. 2021, collected on the Internet by a network security enterprise within its service area.

- We use the traffic data collected in 2017 as the training set and the rest as the testing set.

TABLE X: The F1 score of each method on the real-world dataset.

| Training size | Noise Ratio | FS | Co+FS | DT+FS | ODDS+FS | SMOTE+FS | DT+ODDS+FS | ETA | DT+ODDS+ETA | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| 250 | 20% | .14 ± .20 | .05 ± .16 | .32 ± .28 | .38 ± .19 | .17 ± .06 | .31 ± .25 | .73 ± .28 | .78 ± .30 | .72 ± .04 |
| | 25% | .30 ± .17 | .08 ± .25 | .26 ± .22 | .23 ± .22 | .16 ± .04 | .22 ± .18 | .49 ± .26 | .65 ± .27 | **.70** ± .05 |
| | 30% | .24 ± .13 | .07 ± .20 | .23 ± .15 | .29 ± .15 | .18 ± .04 | .23 ± .13 | .64 ± .34 | .67 ± .30 | **.71** ± .06 |
| | 35% | .23 ± .13 | .25 ± .31 | .24 ± .15 | .25 ± .15 | .18 ± .02 | .21 ± .16 | .47 ± .39 | .49 ± .38 | **.67** ± .04 |
| | 40% | .23 ± .12 | .16 ± .23 | .18 ± .13 | .20 ± .11 | .16 ± .03 | .22 ± .13 | .54 ± .33 | .60 ± .35 | **.86** ± .04 |
| | 45% | .20 ± .09 | .15 ± .11 | .22 ± .09 | .18 ± .09 | .16 ± .02 | .21 ± .09 | .38 ± .32 | .45 ± .35 | **.70** ± .08 |
| 500 | 20% | .15 ± .18 | .15 ± .15 | .22 ± .14 | .15 ± .16 | .15 ± .05 | .37 ± .25 | .68 ± .39 | .68 ± .39 | **.72** ± .05 |
| | 25% | .22 ± .19 | .20 ± .36 | .15 ± .18 | .11 ± .10 | .13 ± .06 | .16 ± .20 | .51 ± .30 | .58 ± .31 | **.76** ± .06 |
| | 30% | .24 ± .17 | .11 ± .28 | .24 ± .19 | .23 ± .16 | .13 ± .05 | .25 ± .18 | .48 ± .30 | .58 ± .28 | **.75** ± .04 |
| | 35% | .20 ± .14 | .19 ± .38 | .25 ± .18 | .19 ± .20 | .15 ± .03 | .19 ± .16 | .31 ± .34 | .47 ± .32 | **.71** ± .02 |
| | 40% | .18 ± .09 | .17 ± .29 | .19 ± .14 | .28 ± .19 | .11 ± .06 | .18 ± .16 | .36 ± .33 | .45 ± .35 | **.76** ± .04 |
| | 45% | .20 ± .09 | .12 ± .15 | .17 ± .10 | .17 ± .09 | .13 ± .05 | .17 ± .11 | .16 ± .25 | .17 ± .27 | **.77** ± .06 |
| 1000 | 20% | .30 ± .27 | .00 ± .01 | .29 ± .34 | .27 ± .23 | .13 ± .06 | .24 ± .21 | .73 ± .31 | .66 ± .28 | **.81** ± .02 |
| | 25% | .36 ± .21 | .00 ± .00 | .29 ± .25 | .23 ± .20 | .15 ± .04 | .21 ± .21 | .55 ± .27 | .44 ± .26 | **.64** ± .05 |
| | 30% | .13 ± .15 | .32 ± .46 | .18 ± .15 | .12 ± .16 | .12 ± .05 | .16 ± .14 | .68 ± .34 | .43 ± .33 | .61 ± .01 |
| | 35% | .20 ± .15 | .41 ± .42 | .28 ± .21 | .25 ± .16 | .12 ± .05 | .30 ± .19 | .50 ± .35 | .45 ± .34 | **.69** ± .04 |
| | 40% | .29 ± .14 | .19 ± .30 | .32 ± .21 | .24 ± .18 | .15 ± .01 | .22 ± .18 | .35 ± .33 | .42 ± .37 | **.63** ± .02 |
| | 45% | .14 ± .09 | .04 ± .07 | .22 ± .16 | .17 ± .10 | .14 ± .02 | .17 ± .09 | .25 ± .26 | .35 ± .30 | **.68** ± .02 |

Our system improves baselines from **89.2% to 445.5%, at an average of 272.5%.**

# Discussion

- Extreme Label Noise (noise ratio ≥ 50%)
  - When the noise ratio is 90%, our system still get an F1 of 0.447 (baseline F1 ≤ 0.001).
  - Extreme noise can be preprocessed to less than 50%. (e.g., filtering malicious traffic through Alexa Top List, and only preserve the normal traffic communicated with well-known website)

- Training Overhead
  - Our system can be trained fast. (with 1000 samples, 3 modules' time are 4032s, 7.5s, and 156s)

- Long-term Deployment
  - Our model can be efficiently re-trained to adapt new normal traffic (due to small overhead)
  - If collected normal samples are diverse, We can cluster them (each cluster will be less diverse) and process each cluster individually.

- Evading Detection
  - Our system has built-in designs of adversarial learning (generate samples that imitate evasion samples), which can handle evading malicious traffic.

# Thanks

qyq21@mails.tsinghua.edu.cn