Front. Comput. Sci., 2024, 18(1): 181809
https://doi.org/10.1007/s11704-023-2386-4

LETTER

MMCo: using multimodal deep learning to detect malicious traffic with noisy labels

Qingjun YUAN[1], Gaopeng GOU[2], Yuefei ZHU[1], Yongjuan WANG (✉)[1]

1 Strategic Support Force Information Engineering University, Zhengzhou 450001, China
2 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

© The Author(s) 2023. This article is published with open access at link.springer.com and journal.hep.com.cn

1 Introduction

The success of a deep learning-based network intrusion detection systems (NIDS) relies on large-scale, labeled, realistic traffic [1,2]. However, automated labeling of realistic traffic, such as by sandbox and rule-based approaches, is prone to errors [3], which in turn affects deep learning-based NIDS.

Several effective schemes for learning with noisy labels (LNL) have been proposed, among which the use of parallel networks for sample selection during training has been shown to be effective. It is argued that parallel networks can discriminate samples from different perspectives, thus adding additional information to the training process. The centerpiece is maintaining disagreement of networks. Both Co-teaching [4] and Co-teaching+ [5] train two networks with the same structure and the same inputs but with different initial weights. Co-learning [6] trains a two-headed encoder that collaboratively performs sample selection and weight updates. The above methods have only a single input, and rely only on different initial network states to maintain disagreement, ignoring multimodalities in the data. However, multimodality is naturally present in the traffic [7,8], and it is difficult to optimize the model using a single modality.

To solve the above problem, we propose MMCo, a **Co**-teaching-like method using **multim**odal information and parallel, heterogeneous networks to detect malicious traffic with noisy labels. Unlike existing methods, (1) MMCo is the first LNL method that uses multimodality to maintain disagreement; and (2) the parallel networks in MMCo are heterogeneous and input different modalities of samples, which can mitigate self-control degradation and enhance robustness.

2 Architecture

The architecture of MMCo is shown in Fig. 1. Multimodal information is extracted from the raw traffic and fed into parallel-trained networks, which perform collaborative training and sample selection from different data perspectives of local trend variation and long-term behavior. Together, the trained networks form a malicious traffic classifier.

Notation Let $\mathbf{D} := (x_a, x_b, \hat{y})$ be a noisy dataset, where x_a and x_b denote the different modalities of the samples and $\hat{y}$ denotes the noisy labels. $\mathcal{F}_{\text{CNN}}$ and $\mathcal{F}_{\text{RNN}}$ refer to the selected CNN and RNN with weight parameters θ_{CNN} and θ_{RNN}, respectively, with corresponding inputs $\mathbf{D}^{(a)} := (x_a, \hat{y})$ and $\mathbf{D}^{(b)} := (x_b, \hat{y})$. For simplicity, we refer to the inputs as $\mathbf{D}$.

Multimodal information extractor We segment the raw traffic and extract different modalities, such as semantic and spatio-temporal modality. In the semantic modality, metadata of the protocol stack and the agreement of the encryption are included, since encrypted packets are highly structured. These metadata are unencrypted, can reflect the communication setup and negotiation process for encryption suites and extensions, and can help identify individuals. Then, we extract the semantic embedding of these fields by a projection network. Spatio-temporal modalities include size sequences and arrival time sequences of the packets that characterize the behavior, such as sending short packets at regular intervals may indicate Trojan's keep-alive behavior. Both the above modalities can help identify malicious traffic from different perspectives. We input them into suitable networks for training with noisy labels.

Parallel training Algorithm 1 represents the training process of MMCo. In this stage, we choose CNN and RNN to learn two different modalities respectively. The semantic modality consists of the control and exchange information represented in the packets, which are aligned and suitable for processing using CNNs. While spatio-temporal modality is essentially two time-series with variable length that RNNs are good at handling.

In each mini-batch, $\mathcal{F}_{\text{CNN}}$ and $\mathcal{F}_{\text{RNN}}$ are fed with different modalities of the same subset. $\mathcal{F}_{\text{CNN}}$ and $\mathcal{F}_{\text{RNN}}$ select for each other the samples they consider more important, i.e., the samples with different distinguish or less loss among all mini-batches. Only these samples will be used for updating the parameters of the networks.

$\mathcal{L}$ is used to estimate the loss of the samples. In steps 5 and 6, the samples with less loss in each mini-batch are selected.

Received June 23, 2022; accepted April 11, 2023

E-mail: pinkywyj@163.com

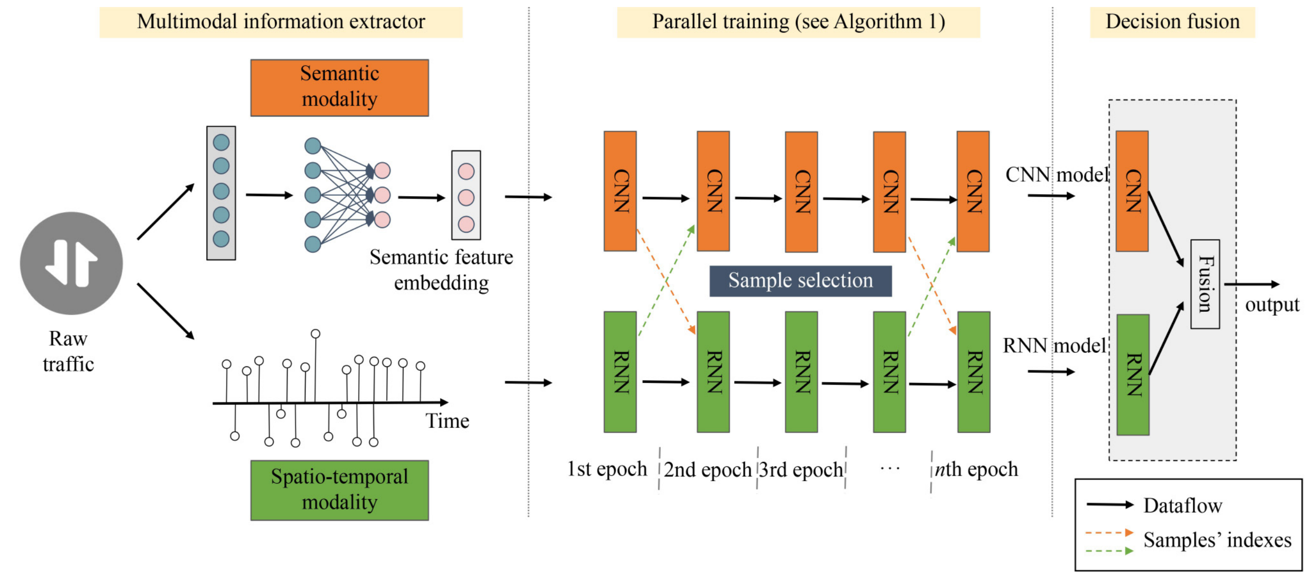

Fig. 1 Architecture of MMCo

```
Algorithm 1 Parallel training in MMCo
Input:  $\theta_{CNN}$, $\theta_{RNN}$, learning rate $\eta$, batch size $Bs$,
    $\lambda$, $\mathcal{L}$.
 1: for $e$ in (1,2,...,$epochs$) do
 2:     Shuffle training set $\mathbf{D}$
 3:     for $n$ in (1,2,...,$|\mathbf{D}|/Bs$ ) do
 4:         Fetch $n$-th mini-batch $\mathbf{D}^*$ from $\mathbf{D}$
 5:         $\mathbf{D}^*_{CNN} \leftarrow \arg\min_{\mathbf{D}^*:|\mathbf{D}^*|\geq\lambda(e)Bs}\mathcal{L}(\mathbf{D}^*;\theta_{CNN})$
 6:         $\mathbf{D}^*_{RNN} \leftarrow \arg\min_{\mathbf{D}^*:|\mathbf{D}^*|\geq\lambda(e)Bs}\mathcal{L}(\mathbf{D}^*;\theta_{RNN})$
 7:         $\theta_{CNN} \leftarrow \theta_{CNN} - \eta\nabla\mathcal{L}\left(\mathbf{D}^*_{RNN};\theta_{CNN}\right)$
 8:         $\theta_{RNN} \leftarrow \theta_{RNN} - \eta\nabla\mathcal{L}\left(\mathbf{D}^*_{CNN};\theta_{RNN}\right)$
 9:     end for
10: end for
11: return $\theta_{CNN}$,$\theta_{RNN}$
```

In steps 7 and 8, θ_{CNN} and θ_{RNN} are updated using the subsets selected by each other, respectively. The relative entropy between the observed and predicted labels is measured.

$$\mathcal{L}(\mathbf{D};\theta) = -\frac{1}{|\mathbf{D}|}\sum_{\mathbf{D}}\hat{y}\log(\mathcal{F}(x;\theta)). \tag{1}$$

λ determines how many samples are selected in each epoch to update θ_{CNN} and θ_{RNN}. Benefiting from the memory properties of neural networks, they can learn the correct knowledge preferentially. As the network continually fits the noisy data distribution, the impact of noisy labels becomes increasingly significant. Therefore, λ is decreased monotonically with *epoch*, and the range in this paper is [0.5,0.9].

Decision fusion In the decision fusion stage, we used the classical late classifier fusion, i.e., constructing a weighted linear combination of the scores of both classifiers.

$$H(x) = w_{\text{CNN}}\mathcal{F}(x;\theta_{\text{CNN}}) + w_{\text{RNN}}\mathcal{F}(x;\theta_{\text{RNN}}). \tag{2}$$

3 Experiment and analysis

Datasets We need to extract multimodal information from raw traffic. Therefore, we choose the pcaps provided by CICIDS-2017 [9] and DoHBrw-2020 [10], which are divided into training and validation sets. The labels in the training set are flipped randomly as noise. We set up **Sym**metric and **Asym**metric scenarios according to realistic situations. In the symmetric scenarios, all the labels are likely to be flipped, while in the asymmetric scenarios, only some classes are flipped. In the validation set, all labels remain unchanged.

Results The disagreement of the two networks is shown in Fig. 2. The accuracy on the validation set is shown in Fig. 3. When 200 epochs are completed, the classification networks of MMCo still maintain 10% disagreement with a final classification accuracy of 90%, while the disagreement of the two networks of other methods is close to zero. At this time, the two networks are in a state of self-control degradation, and it is difficult to learn more knowledge. However, MMCo can maintain a higher disagreement compared to others, thus

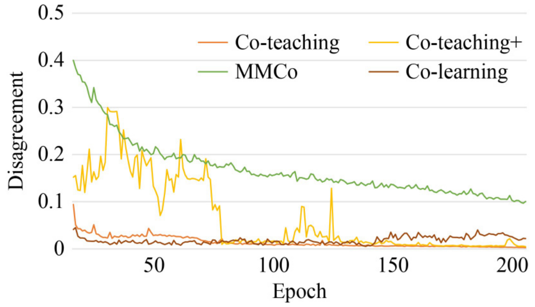

Fig. 2 Disagreement of networks (Sym-20%)

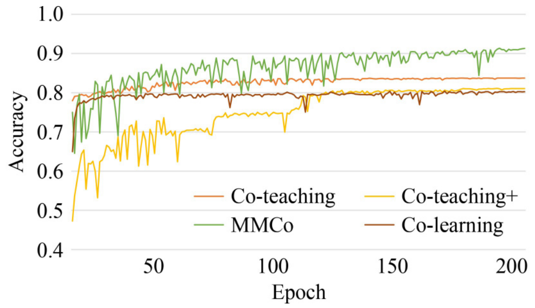

Fig. 3 Accuracy on validation set (Sym-20%)

helping the classifiers to learn more correct knowledge, with about 10% higher accuracy.

Table 1 compares MMCo and other methods under different patterns and noise levels, and MMCo outperforms state-of-the-art methods.

Table 1 Accuracy under different noise scenarios

Acc(%)	Asym.			Sym.	
	20%	50%	70%	20%	40%
Co-teaching	83.13	81.17	73.55	83.61	73.11
Co-teaching+	80.49	79.47	71.88	79.89	72.83
Co-learning	80.24	80.11	74.71	80.35	75.54
MMCo (ours)	**92.89**	**90.76**	**85.31**	**92.86**	**81.31**

4 Conclusion and future work

In this paper, we validated the feasibility of LNL using multimodal information. We found that MMCo could maintain higher disagreement through networks with different structures and modal inputs. This improved the robustness of the classifier. Thus, MMCo could alleviate the problem of self-control degradation of parallel models, to which current LNL methods are prone.

In the future, we will further investigate the analysis of the representations of two networks in multimodal networks using explainable artificial intelligence, which may help identify and clean malicious traffic with noisy labels.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Sun X, Ma S, Li Y, Wang D, Li Z, Wang N, Gui G. Enhanced echo-state restricted Boltzmann machines for network traffic prediction. IEEE Internet of Things Journal, 2020, 7(2): 1287–1297
2. Popoola S I, Ande R, Adebisi B, Gui G, Hammoudeh M, Jogunola O. Federated deep learning for zero-day botnet attack detection in IoT-edge devices. IEEE Internet of Things Journal, 2022, 9(5): 3930–3944
3. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Computers & Security, 2019, 86: 147–167
4. Han B, Yao Q, Yu X, Niu G, Xu M, Hu W, Tsang I W, Sugiyama M. Co-teaching: robust training of deep neural networks with extremely noisy labels. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018, 8536−8546
5. Yu X, Han B, Yao J, Niu G, Tsang I W, Sugiyama M. How does disagreement help generalization against label corruption? In: Proceedings of the 36th International Conference on Machine Learning. 2019, 7164−7173
6. Tan C, Xia J, Wu L, Li S Z. Co-learning: learning from noisy labels with self-supervision. In: Proceedings of the 29th ACM International Conference on Multimedia. 2021, 1405−1413
7. Aceto G, Ciuonzo D, Montieri A, Pescapé A. DISTILLER: encrypted traffic classification via multimodal multitask deep learning. Journal of Network and Computer Applications, 2021, 183−184: 102985
8. Nascita A, Montieri A, Aceto G, Ciuonzo D, Persico V, Pescapé A. XAI meets mobile traffic classification: understanding and improving multimodal deep learning architectures. IEEE Transactions on Network and Service Management, 2021, 18(4): 4225–4246
9. Sharafaldin I, Lashkari A H, Ghorbani A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy. 2018, 108−116
10. MontazeriShatoori M, Davidson L, Kaur G, Lashkari A H. Detection of DoH tunnels using time-series classification of encrypted traffic. In: Proceedings of the 5th Cyber Science and Technology Congress. 2020, 63−70