A Lightweight Malware Traffic Classification Method Based on a Broad Learning Architecture

Yibin Zhang, *Student Member, IEEE*, Guan Gui, *Senior Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*—Malware traffic classification (MTC) plays an important role for securing the Internet of Things (IoT). Many machine learning (ML) and deep learning (DL)-based MTC methods have been proposed in recent years. However, the former still requires human intervention, while the latter incurs considerable computation overheads. To address these problems, we propose a broad learning (BL)-aided MTC method (BL-MTC), which is a lightweight and graphics processing unit-free solution with good performance and extremely low cost. The simulation results show that the proposed BL-MTC method not only achieves superior results on the USTC-TFC2016 data set but also exhibits an exponential advantage in computation overhead.**

***Index Terms*—Broad learning (BL), graphics processing unit (GPU)-free, malware traffic classification (MTC), secure Internet of Things (IoT).**

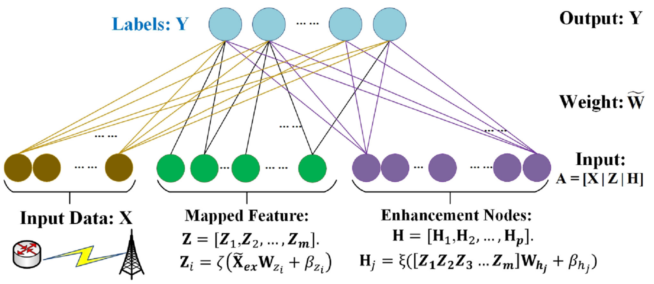
Labels: Y
Output: Y
Weight: $\widetilde{\mathbf{W}}$
Input:
$\mathbf{A} = [\mathbf{X}|\mathbf{Z}|\mathbf{H}]$
Input Data: X
Mapped Feature:
$\mathbf{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_m]$.
$\mathbf{Z}_i = \zeta(\widetilde{\mathbf{X}}_{ex}\mathbf{W}_{z_i} + \beta_{z_i})$
Enhancement Nodes:
$\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_p]$.
$\mathbf{H}_j = \xi([\boldsymbol{Z}_1\boldsymbol{Z}_2\boldsymbol{Z}_3 \dots \boldsymbol{Z}_m]\mathbf{W}_{h_j} + \beta_{h_j})$

Fig. 1. Illustration of the proposed BL-MTC method. It is composed of the input matrix, the optimized weight, and the output labels.

I. Introduction

Under the threat of Corona Virus Disease 2019 (COVID-19), telecommuting has become the best choice for many Internet enterprises. With the rapid development of wireless communications and Internet of Things (IoT), more and more enterprises have paid attention to the application of Industrial IoT (IIoT), which allows their employees to work remotely. However, when people enjoy the convenience of telecommuting, they also need to deal with its security and reliability problems [1]. Hence, the malware traffic classification (MTC) technology plays a significant role in guaranteeing the security and reliability of the Internet and IIoT. Due to the explosive growth of network traffic, it is almost impossible to complete MTC by manual means. In the past decade, many machine learning (ML) or deep learning (DL)-based methods have been proposed for MTC. Unfortunately, both ML-based and DL-based MTC methods have their limitations [2]. The former still requires manually traffic feature extraction, while the latter usually incurs considerable computation overheads. To address these problems, this letter explores a broad learning (BL) [3]-based MTC method (termed BL-MTC), which is a lightweight solution without using a graphics processing unit (GPU). The main contributions of this letter are as follows.

1) This letter proposes an efficient, lightweight, and low-cost MTC method utilizing the BL architecture, called BL-MTC. According to the best of our knowledge, this is the first attempt to apply BL for MTC tasks.
2) The proposed BL-MTC is evaluated on the real USTC-TFC2016 [4] data set, including both benign and malware traffic. Experimental results show that BL-MTC achieves a high classification performance at a lower computation overhead than several existing methods.

II. System Model and Proposed BL-MTC Method

A. System Model

The MTC task is to solve the matching problem between the received network traffic and the corresponding traffic source. Assume that the data set is defined as $\mathbb{D}\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the captured network traffic samples, $\boldsymbol{y}_i$ is the label of the corresponding source, and N is the number of categories. Then, the problem to be solved can be defined as

$$\widehat{\boldsymbol{y}_i} = f_{\mathrm{MTC}}(\boldsymbol{x}_i, \widetilde{\mathbf{W}}), \ i = 1, 2, \dots, N \tag{1}$$

where $f_{\mathrm{MTC}}(\cdot)$ represents the mapping function to classify the received network traffic, $\widehat{\boldsymbol{y}_i}$ is the classification label of $\boldsymbol{x}_i$, and $\widetilde{\mathbf{W}}$ denotes the optimal weights. We define the probability of a group of network traffic being classified correctly as the accuracy rate, given by $\alpha = \sum_{i=1}^{N}(\widehat{\boldsymbol{y}_i} = \boldsymbol{y}_i)/N$, and the corresponding error rate is $\epsilon = 1 - \alpha$. Therefore, we have the objective function for the highest accuracy and the lowest error, i.e.,

$$\widetilde{\mathbf{W}} = \arg\min_{\mathcal{W}} \ \epsilon\left\{\widehat{\boldsymbol{y}_i} = f_{\mathrm{MTC}}(\boldsymbol{x}_i, \mathcal{W}), \boldsymbol{y}_i\right\} \tag{2}$$

where the optimal weight $\widetilde{\mathbf{W}}$ and the framework of $f_{\mathrm{MTC}}(\cdot)$ are the solution to the MTC problem.

B. Our Proposed BL-MTC Method

This letter explores a lightweight and computationally efficient method to solve MTC tasks. The traditional DL-based MTC methods are focused on building an excellent mapping function architecture, i.e., $f_{\mathrm{MTC}}(\cdot)$. Through the forward and back propagation mechanism, the optimal weight $\widetilde{\mathbf{W}}$ is identified through multiple iterations. In contrast, the proposed BL-MTC pays more attention to solving $\widetilde{\mathbf{W}}$ with matrix theory, as shown in Fig. 1. It is obvious that the proposed BL-MTC has the simplest one-layer forward architecture. The MTC

Manuscript received 8 March 2023; revised 25 June 2023; accepted 17 July 2023. Date of publication 20 July 2023; date of current version 21 November 2023. This work was supported in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX22-0948, and in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 22KJA510002. *(Corresponding author: Guan Gui.)*

Yibin Zhang and Guan Gui are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: 2021010208@njupt.edu.cn; guiguan@njupt.edu.cn).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Digital Object Identifier 10.1109/JIOT.2023.3297210

TABLE I
PERFORMANCE AND COMPUTATION OVERHEAD COMPARISON OF DIFFERENT MTC METHODS ON BENIGN TRAFFIC

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Computation Overhead (s)	
					CPU-based	GPU-based
BL-MTC (proposed)	**99.476**	**99.542**	**99.58**	**99.556**	**72.429**	**–**
ResNet34-MTC	99.897	99.896	99.881	99.881	39,752.118	3,378.042
VGG16-MTC	99.992	99.993	99.993	99.993	498,852.045	6,534.655
AlexNet-MTC	99.918	99.925	99.926	99.925	19,109.061	2,773.089
CNN-MTC	99.985	99.987	99.986	99.986	18,971.371	2,083.418

TABLE II
PERFORMANCE AND COMPUTATION OVERHEAD COMPARISON OF DIFFERENT MTC METHODS ON MALWARE TRAFFIC

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Computation Overhead (s)	
					CPU-based	GPU-based
BL-MTC (proposed)	**98.672**	**98.748**	**98.737**	**98.74**	**70.932**	**–**
ResNet34-MTC	99.166	99.162	99.173	99.167	36,157.524	3669.742
VGG16-MTC	99.671	99.671	99.670	99.670	434,072.377	6,922.176
AlexNet-MTC	98.381	98.397	98.415	98.404	17198.291	1813.45
CNN-MTC	98.945	98.947	98.942	98.944	17740.461	1288.729

task is modeled as a matrix theory problem, given by

$$\mathbf{Y} = \mathbf{A} \cdot \widetilde{\mathbf{W}} = [\mathbf{X}|\mathbf{Z}|\mathbf{H}] \cdot \widetilde{\mathbf{W}}, \tag{3}$$

where $\mathbf{A}$ is composed of the row traffic data $\mathbf{X}$, the mapped feature nodes $\mathbf{Z}$, and the enhancement nodes $\mathbf{H}$. The mapped feature nodes and the enhancement nodes are linear and nonlinear feature nodes, respectively, given by

$$\mathbf{Z}_i = \zeta\left(\tilde{\mathbf{X}}_{ex}\mathbf{W}_{z_i} + \boldsymbol{\beta}_{z_i}\right),\ i = 1, 2, \ldots, m, \tag{4}$$

$$\mathbf{H}_j = \xi\left(\left[\mathbf{Z}_1\mathbf{Z}_2, \ldots \mathbf{Z}_m\right]\mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j}\right),\ j = 1, 2, \ldots, p \tag{5}$$

where $\zeta(\cdot)$ and $\xi(\cdot)$ are linear and nonlinear transformation functions, respectively, $\mathbf{W}_{z_i}$ and $\mathbf{W}_{h_j}$ are randomly initialized weights, and β_{z_i} and β_{h_j} are randomly initialized biases. The key idea of BL-MTC is to solve for the optimized $\widetilde{\mathbf{W}}$ in (3) through measured traffic data $\mathbf{X}$ and label $\mathbf{Y}$, given by

$$\widetilde{\mathbf{W}} = \mathbf{A}^{-1} \cdot \mathbf{Y} \approx \mathbf{A}^{\dagger} \cdot \mathbf{Y}. \tag{6}$$

Note that the input matrix $\mathbf{A}$ is huge and it is difficult to find the corresponding inverse matrix $\mathbf{A}^{-1}$. Therefore, we use the pseudo-inverse operation to obtain $\mathbf{A}^{\dagger} = \lim_{\lambda \to 0}(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T$, to effectively solve for the optimal weight $\widetilde{\mathbf{W}}$.

III. EXPERIMENTAL RESULTS

A. Data Set and Experimental Setup

1) Data Set: The USTC-TFC2016 data set contains ten types of benign traffic, including BitTorrent, Facetime, FTP, Gmail, MyDQL, Outlook, Skype, SMB, Weibo, and WorldOfWarcraft. It also includes ten types of malware traffic captured from real environment, including Cridex, Geodo, Htbot, Miuref, Neris, Nsis-ay, Shifu, Tinba, Virut, and Zeus.

2) Simulation Platform: All the simulations are carried out on a workstation equipped with two Intel Xeon Silver 4210R CPUs and four Nvidia RTX 2080Ti GPUs.

B. Experimental Results

The experimental results are shown in Tables I and II. Correspondingly, Table I shows the classification performance of benign traffic, and Table II shows that of malware traffic. We evaluate BL-MTC using five indicators: 1) accuracy; 2) precision; 3) recall; 4) F1 score; and 5) computation overhead. The benchmarks choice for comparison is ResNet34, VGG16, AlexNet, and CNN. It should be noted that the proposed BL-MTC method does not require testing on the GPU platform. The key idea of BL is to solve matrix equations, hence, the calculation tasks of BL are all matrix operations, which do not require GPU acceleration. The results show that the proposed BL-MTC achieves a great performance compared with the benchmarks. However, it is worth mentioning that the computation overhead of BL-MTC exhibits exponential advantages.

IV. CONCLUSION

The proposed BL-MTC method was proved as a lightweight and GPU-free solution with a satisfactory network traffic classification performance. This is the first application of the BL architecture to MTC. The proposed BL-MTC provided a feasible traffic detection scheme for IIoT edge gateway devices with limited computational power. It aimed to help further improve the security of IIoT. Meanwhile, we proved the feasibility of BL-MTC for CPU platforms, which is a unique and desirable advantage over the DL-based MTC methods. In the next stage, we will focus on the hardware deployment of BL-MTC for industrial applications. We hope to explore BL technology to solve the problem of DL dependence on GPUs, aiming at solving the application shortcomings.

REFERENCES

[1] Z. M. Fadlullah, B. Mao, and N. Kato, "Balancing QoS and security in the edge: Existing practices, challenges, and 6G opportunities with machine learning," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2419–2448, 4th Quart., 2022.

[2] J. Ning et al., "Malware traffic classification using domain adaptation and ladder network for secure Industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17058–17069, Sep. 2022.

[3] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[4] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. ICOIN*, Da Nang, Vietnam, 2017, pp. 712–717.