BADRES: REVEAL THE BACKDOORS THROUGH RESIDUAL CONNECTION

Mingrui He1 3, Tianyu Chen1 3, Haoyi Zhou1 3, Shanghang Zhang2, Jianxin Li1 3

1BDBC, Beihang University, China 2Peking University, China 3SKLSDE, Beihang University

ABSTRACT

Generally, residual connections are indispensable network components in building Convolutional Neural Networks (CNNs) and Transformers for various downstream tasks in Computer Vision (CV), which encourages skip/short cuts between network blocks. However, the layer-by-layer loopback residual connections may also hurt the model’s robustness by allowing unsuspecting input. In this paper, we proposed a simple yet strong backdoor attack method called BadRes, where the residual connections play as a turnstile to be deterministic on clean inputs while unpredictable on poisoned ones. We have performed empirical evaluations on four datasets with ViT and BEiT models, and the BadRes achieves 97% attack success rate without any performance degradation on clean data. Moreover, we analyze BadRes with state-of-the-art defense methods and reveal the fundamental weakness lying in residual connections.

Index Terms— Backdoor attack, neural networks, residual connection

1. INTRODUCTION

Deep neural networks (DNNs) have been widely adopted in many areas. Studies [1, 2] show that network depth is very important in DNNs, and deeper networks tends to perform better. However, deep networks may suffer from network degradation during the training process. To overcome this limitation, residual connection [3] is proposed. By creating a "shortcut" between input and output, residual connection prevents the degradation problem and make it possible to stack more layers into DNN. Despite its widespread success in CV [4, 5] and NLP models [6, 7], residual connection may also provide convenience for malicious attackers.

One of the stealthy and harmful attacks threatening the security of DNNs is backdoor attack [8]. Attackers often set up backdoor by poisoning data [9, 10] or adding backdoor modules [11] during training phase. When inference with benign inputs, the victim model performs normally. Once fed with designed trigger, the victim model would be controlled by the attacker.

The corresponding author is Jianxin Li. This work is supported by the NSFC through grant No.62225202.

The backdoor attack can use residual connection as a shortcut, as shown in Figure 1. When input with a designed trigger, the residual connection of the victim model allows the trigger to survive a stack of neural blocks and be uncovered by the final prediction layer. Furthermore, the forward flow of the features is also controlled by residual connection, which opens a new stealthy way to embed backdoor in victim models. Instead of poisoning the labels [9] or adding backdoor modules [11], the malicious attackers may target the forward flow. The backdoor could mislead the features flow into one designed shortcut, thus directly threatening the final prediction layer. To demonstrate the server risk on popular models based on residual connections, we propose BadRes, the first backdoor attack targeting the residual connections.

Although there are several defense methods for backdoor attacks like purifying the input [12], purifying the model [13, 14] and detecting backdoor [15, 16]. The current backdoor defense methods are designed for detecting poisoned weights rather than defending against the flow change of the model, which are hard to eliminate the threats caused by BadRes.

In this paper, we experiment with three popular residual-based models across four visual datasets. The results reveal that our attack method achieves 97% attack success rate while sacrificing less than 1% accuracy on clean data, outperforming all state-of-the-art backdoor attack methods. Meanwhile, we adopt one defense method against BadRes. Experimental results show that BadRes is more stealthy and hard to eliminate. We also analyze the behavior of the victim models under BadRes threats and reveal their fundamental structure weak-
ness.

2. METHODOLOGY

In this section, we will present our proposed method. Before that, we would like to review the definition related to backdoor attacks.

2.1. Backdoor Preliminary

As mentioned in the previous section, a backdoor attack can be defined as follows. Given a benign input $x_i$, the prediction result $z_i$ of the backdoor model $F'$ is consistent with the ground truth $y_i$ with high probability. In this case, the performance of the backdoor model $F'$ is the same as the clean model $F$. On the other hand, given the poisoned input $x'_i$, the predicted result $z'_i = F'(x'_i)$ will always be the target class $Y'_i$ set by the attacker. Next, we will specify the capabilities and targets of the backdoor attacker.

**Attacker’s capabilities.** We assume that the attacker is able to have all privileges (modify the model structure, training method, and training results) before making it available to the user [8]. During the model inference process, the attacker cannot manipulate the model. This threat scenario can occur in many realistic situations, such as when third parties provide models and interfaces. Correspondingly, the defender can add defense modules or sanitize the model after obtaining the model source file.

**Attacker’s goals.** In general, the attacker aims for the effectiveness and stealthiness of the backdoor attack. Effectiveness refers to the attacker’s ability to control the model output through the backdoor. Stealthiness means that the backdoor is not easily detected, and the backdoor attack is still effective against the mainstream defense.

2.2. BadRes

The workflow of BadRes consists of three steps: the poisoned model phase, the poisoned data phase, and the training phase. The details will be described below.

**Poisoning model phase.** In this phase, the backdoor attacker strengthens the model’s ability to extract backdoor features by modifying the one-layer residual connection structure to a BadRes block.

Before introducing the BadRes block, we first review the residual network structure. In order to solve the degradation problem of deep network, the authors use shortcut connections to transmit the input $x$ to the output as the initial result.

The goal of training is to make the $F(x)$ result approach zero. Compare with not introducing residuals, this approach can highlight minor changes more and is easier to fit residuals. However, from the perspective of backdoor attack, if the input $x$ contains error information, the shortcut connection will save such error and pass it to the deeper network layer, resulting in the output of the network layer being misleading as well afterwards. Based on this characteristic, we design the BadRes block, as shown in Figure 2. The target output $H^*(x)$ is:

$$H^*(x) = F^*(x) - \alpha x$$  \hspace{1cm} (1)

The $\alpha$ represents the poison coefficient. In this way, $F(x)$ becomes $F^*(x) = H^*(x) + \alpha x$. It also makes it easier for $F^*(x)$ to learn the trigger features in poisoned sample $x$. This approach enables the model to learn the trigger features more quickly without changing the feature distribution of the benign inputs, and thus is not easily detected by existing backdoor detection methods, as analyzed in the experiments section.

**Poisoning data phase.** We follow the setup with other backdoor attack methods, where the attacker adds the trigger to part of the benign training dataset and modifies the corresponding label as the target label. We set $D$ to represent all training data sets, and take data from $D$ as poisoned samples in a certain proportion, which is called poison rate $\gamma$. The specific formula is $\gamma = \frac{D_p}{D}$, where $D_p$ is the set of poisoned samples with the trigger added. For the poisoned sample $x'$, the target class is usually set to $y_t$, which is generated as shown in Equation 2:

$$x' = (1 - \lambda)x + \lambda t$$  \hspace{1cm} (2)

The $t$ in the equation represents the trigger. Due to this way of pasting the trigger on the original input, the residual connection will preserve the trigger features, while the BadRes block will take advantage of the shortcut connection to amplify the trigger features.

**Training phase.** During the training process, one layer of the residual connection is selected to be changed to BadRes block, and the rest of the network is not modified. For our poisoned model parameter $\theta'$, the objective is to solve the following optimization problem:

$$\theta' = \arg \min_\theta E_{x \sim X,x' \sim X'} [\mathcal{L}(x,y,\theta) + \mathcal{L}(x',y',\theta')]$$  \hspace{1cm} (3)

Where $x$ and $y$ are benign input and benign target output, respectively, $x'$ is the poisoned sample. $y'$ is the target label, and $\mathcal{L}$ means the loss function. The backdoor attack of BadRes is completed when the training is finished. In the inference phase, the model suffered from BadRes attack would perform normally on the benign test sample, and the output would be the target label after adding the backdoor trigger.

3. EXPERIMENTAL RESULTS AND DISCUSSION

3.1. Experimental settings

We conducted experiments on four datasets, including CIFAR-100[17], Food-101[18], MNIST[19] and ImageNet[20].
Fig. 2. Illustration of the residual backdoor process for the poisoning layer indexed as k. For the poisoned input with the trigger added, the image which is originally a train is identified as a rocket after the Badres block. The red part in the Transformer encoder is the badres block.

MNIST is a 10-classification dataset with black and white images, containing 49K training data and 1K test data. CIFAR-100 has 100 classes, including 50K training data and 10K test data. Food-101 has 101 classes with 1K images per category, of which 75,750 images are used for training and the rest for testing. For the ImageNet dataset, to simplify the testing, we randomly select 10K of the data for training and 1K for testing.

For the victim models, we choose three self-attention models that work well in CV tasks and use residual connections for our experiments. They are ViT[5], DeiT[21] and BEiT[22]. For all victim models, the patch size is $16 \times 16$.

We compare BadRes with two backdoor attack methods. One is the classic backdoor attack method BadNets[9], and the other is the Blended attack[23]. Meanwhile, we choose STRIP[12] to test the stealthiness of the three attack methods.

For the experimental setup of the backdoor attack, we choose a poisoning rate of 10% for all the backdoor attack datasets and the target labels are zero. All images are resized to $224 \times 224$. A total of four different bicolor grid maps are tested as triggers. In our BadRes method, the hyperparameter $\alpha$ in the BadRes block is 0.5. For the BadNets and BadRes methods, the poisoning samples are generated by pasting a trigger of size $16 \times 16$ in the lower right corner of the image. For the Blended method, the poisoned samples are generated by blending the triggers with the whole image, where the transparency of the triggers is 5%. For the training process of all backdoor attacks, we set the initial learning rate to 0.0001, the batch size to 32, the epoch size to 30, and use the Adam optimizer[24].

We use two metrics that are consistent with BadNets[9]. For the case where the input does not add the trigger, we use the benign accuracy (BA) to evaluate the different attacks. For the case of adding triggers to the inputs, we use the attack success rate (ASR) to evaluate the attacks, which is the rate of successfully misleading the model output to the target class among all the poisoned samples.

### Table 1. Comparison of different methods against Vision Transformer without defense. Among all attacks, the best result is denoted in boldface.

| Method | BadRes | BadNets | Blended |
|--------|--------|---------|---------|
| MNIST  | 97.67  | 99.42   | 99.74   |
| CIFAR-100 | 95.96 | 92.37   | 95.57   |
| Food-101 | 98.57 | 83.43   | 86.80   |
| ImageNet | 100   | 90.67   | 98.24   |
| Average | 98.05 | 91.47   | 97.11   |

#### 3.2. Attack effectiveness

We follow the experimental setup in the previous section to experiment. In order to fairly and comprehensively evaluate the effectiveness of BadRes, we test it separately for different datasets with different victimization models.

Firstly, the detailed results of the experiments for different datasets are shown in Table 1. We use three backdoor attacks for the ViT model, and the experimental results are the average of four different triggers. As the image pixels increase, the ASR of the BadNets method gradually starts to decrease. We analyze that this is because, at the higher pixel level, this pasted trigger is in a small percentage and may only affect part of the patch embeddings for ViT. This makes the model unable to learn the features of the target label during the forward propagation, and the original correct feature extraction is also affected. In contrast, BadRes and Blended are more stable than BadNets, and BadRes is better in overall average ASR and BA. On the ImageNet dataset, the BA of BadRes can be 2 percentage points higher than Blended, which can also show that our proposed method has almost no effect on the classification results of benign data.

Secondly, we experiment on the imageNet dataset for
three different models, and the results are shown in Table 2. For the ViT model, the results are consistent with the above experimental results on different datasets, and the BadRes and Blended methods are more effective than BadNets. For the BEiT model, we find that Blended works best and the ASR reaches 93.75%. We analyze that this is because BEiT uses the mask mechanism for pre-training, which can defend against pasted backdoor attacks to a certain extent. Our BadRes method is still able to reach 90% ASR with the same triggers as BadNets, which shows that the BadRes method successfully strengthens the backdoor injection of error messages. For DeiT using the distillation mechanism, the attack effectiveness of both BadNets and Blended decreases to a certain extent, while the BadRes’ ASR can still reach 100%. This indicates that our proposed method is hardly affected by distillation. Overall, for the three different victimization models, our proposed method is more threatening and has less impact on the performance of the original model.

### 3.3. Attack stealthiness

![Fig. 3. Entropy of different backdoor attack methods generated based on STRIP[12] method. Higher Entropy means harder to be detected that the model suffers from backdoor attack.](image)

We choose the STRIP method to test the stealthiness of different backdoor attack methods, which detects the backdoor attack by adding perturbations to the input and then calculating the average entropy of the predictions, where the lower entropy indicates that the model is more likely to have been injected with a backdoor. We choose the ViT model to calculate the entropy after different attacks on two datasets, MNIST and CIFAR-100, and the results are shown in Figure 3. The entropy of our proposed method on both datasets is much higher than that of BadNets and Blended, which also indicates that BadRes is more difficult to be detected by this defense method and has better stealthiness.

### 3.4. Ablation studies

#### The selection of poisoned layer index

We conduct experiments on both ImageNet and Food-101 datasets, and the hyperparameters are consistent with Section 3.1 except that the poisoned layer index is changed. When the layer index is less than or equal to nine, the ASR is relatively stable and close to 100%. When the layer index is the last two layers, the ASR drops sharply, especially when the last layer is selected, the ASR drops to below 60% in the ImageNet dataset. This phenomenon demonstrates that when the index of the poisoned residual unit is small, it can pass down the wrong information and produce a domino effect, thus achieving a higher ASR. On the other hand, the BA rose as the poisoning layer index increases. This also indicates that the larger the poisoning layer index in the case of clean data, the less the impact of the model’s ability to extract the correct features.

#### Effect of poisoning rate

To discuss the effect of the poisoning rate on BadRes, consistent with the previous section, we conduct experiments on both datasets and varied only the poisoning rate. The ASR exceeds 90% when the poisoning rate reaches 0.05. This result shows that BadRes can learn the features of triggers with only a small amount of poisoning data. On the other hand, the BA does not decrease much with the increase of poisoning rate, stabilizing at around 90% and 84% on the ImageNet and food101 datasets. This also indicates that our proposed method has good stability.

### 4. CONCLUSION AND FUTURE WORKS

In this paper, we propose BadRes, a backdoor attack method against residual connections. We test the effectiveness of BadRes on four popular datasets and three mainstream victim models, and it outperforms BadNets and Blended in terms of average ASR and BA. Meanwhile, we use STRIP to test the stealthiness of our method, and the results demonstrate that BadRes is more difficult to be detected and eliminated compared to the other two baseline models. We then use ablation experiments to verify the robustness of BadRes.

Finally, we hope that our work will bring attention to the security of residual connections and that practitioners should be aware of the potential risks when applying this technique. On the other hand, how to improve the residual connection structure to resist such a backdoor attack deserves further research.
5. REFERENCES

[1] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR 2015., 2015.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, et al., “Going deeper with convolutions,” in IEEE CVPR 2015. 2015, pp. 1–9, IEEE Computer Society.

[3] Abhishek Verma, Hussam Qassim, and David Feinheimer, “Residual squeeze CNDS deep learning CNN model for very large scale places image recognition,” in 8th IEEE UEMCON 2017. 2017, pp. 463–469, IEEE.

[4] Fei Wang, Mengqing Jiang, Chen Qian, et al., “Residual attention network for image classification,” in IEEE CVPR 2017. pp. 6450–6458, IEEE Computer Society.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in ICLR. 2021, OpenReview.net.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., “Attention is all you need,” in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 2017, pp. 5998–6008.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in NAACL-HLT 2019. 2019, pp. 4171–4186, Association for Computational Linguistics.

[8] Yiming Li, Baoyuan Wu, Yong Jiang, et al., “Backdoor learning: A survey,” CoRR, vol. abs/2007.08745, 2020.

[9] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain,” CoRR, vol. abs/1708.06733, 2017.

[10] Yuezun Li, Yiming Li, Baoyuan Wu, et al., “Invisible backdoor attack with sample-specific triggers,” in ICCV. 2021, pp. 16443–16452, IEEE.

[11] Yingqi Liu, Shiqiang Ma, Yousra Aafer, et al., “Trojaining attack on neural networks,” in NDSS. 2018, The Internet Society.

[12] Yansong Gao, Change Xu, Derui Wang, et al., “STRIP: a defence against trojan attacks on deep neural networks,” in ACSAC. 2019, pp. 113–125, ACM.

[13] Bolun Wang, Yuanshun Yao, Shawn Shan, et al., “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” in IEEE SP 2019. 2019, pp. 707–723, IEEE.

[14] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, “Fine-pruning: Defending against backdooring attacks on deep neural networks,” in RAID. 2018, vol. 11050 of Lecture Notes in Computer Science, pp. 273–294, Springer.

[15] Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, et al., “T-miner: A generative approach to defend against trojan attacks on dnn-based text classification,” in 30th USENIX Security Symposium, USENIX Security 2021. 2021, pp. 2255–2272, USENIX Association.

[16] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar, “Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks,” in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019. 2019, pp. 4658–4664, ijcai.org.

[17] Alex Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

[18] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool, “Food-101 - mining discriminative components with random forests,” in Computer Vision - ECCV 2014 - 13th European Conference, Zurich. 2014, vol. 8694 of Lecture Notes in Computer Science, pp. 446–461, Springer.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] Olga Russakovsky, Jia Deng, Hao Su, et al., “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al., “Training data-efficient image transformers & distillation through attention,” in ICML. 2021, vol. 139 of Proceedings of Machine Learning Research, pp. 10347–10357, PMLR.

[22] Hangbo Bao, Li Dong, and Furu Wei, “Beit: BERT pre-training of image transformers,” CoRR, vol. abs/2106.08254, 2021.

[23] Xinyun Chen, Chang Liu, Bo Li, et al., “Targeted backdoor attacks on deep learning systems using data poisoning,” CoRR, vol. abs/1712.05526, 2017.

[24] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR, 2015.