BadScan: An Architectural Backdoor Attack on Visual State Space Models

Om Suhas Deshmukh
IIT Dharwad
210010033@iitdh.ac.in

Sankalp Nagaonkar
IIT Dharwad
210020031@iitdh.ac.in

Achyut Mani Tripathi
IIT Dharwad
t.achyut@iitdh.ac.in

Ashish Mishra
HPE lab, Bangalore
mishraashish632@gmail.com

Abstract

The newly introduced Visual State Space Model (VMamba), which employs State Space Mechanisms (SSM) to interpret images as sequences of patches, has shown exceptional performance compared to Vision Transformers (ViT) across various computer vision tasks. However, recent studies have highlighted that deep models are susceptible to adversarial attacks. One common approach is to embed a trigger in the training data to retrain the model, causing it to misclassify data samples into a target class, a phenomenon known as a backdoor attack. In this paper, we first evaluate the robustness of the VMamba model against existing backdoor attacks. Based on this evaluation, we introduce a novel architectural backdoor attack, termed BadScan, designed to deceive the VMamba model. This attack utilizes bit plane slicing to create visually imperceptible backdoored images. During testing, if a trigger is detected by performing XOR operations between the k^{th} bit planes of the modified triggered patches, the traditional 2D selective scan (SS2D) mechanism in the visual state space (VSS) block of VMamba is replaced with our newly designed BadScan block, which incorporates four newly developed scanning patterns. We demonstrate that the BadScan backdoor attack represents a significant threat to visual state space models and remains effective even after complete retraining from scratch. Experimental results on two widely used image classification datasets, CIFAR-10, and ImageNet-1K, reveal that while visual state space models generally exhibit robustness against current backdoor attacks, the BadScan attack is particularly effective, achieving a higher Triggered Accuracy Ratio (TAR) in misleading the VMamba model and its variants.

1. Introduction

Convolutional neural networks (CNNs) such as ResNet [13], Inception [29], and EfficientNet [15], along with Transformer-based models like ViT [4], Conformer [25], Resformer [30], and MLP-mixer [31], have been widely applied to a range of computer vision tasks. These include object detection, image classification, anomaly detection, segmentation, and multimodal learning tasks. These models have consistently delivered state-of-the-art performance across various domains. The recently proposed Mamba models [10, 32], inspired by state-space models (SSMs) [17], have outperformed existing deep learning models and established new benchmarks in various NLP tasks. This success has generated interest in the computer vision community to investigate SSMs for visual tasks [43]. In [43], the first visual state-space model VMamba was introduced, featuring a 2D selective scan (SS2D)-based visual state space (VSS) block that analyzes image patches in four directions to improve feature extraction. The VMamba [43] model and its variants [3], [22], [37], [40] have achieved state-of-the-art performance in numerous computer vision tasks, emerging as strong competitors to transformer-based models like ViTs. Unlike transformer-based models, which suffer from quadratic complexity, the VMamba model offers linear computational efficiency.

Despite their exceptional performance across various domains, the deep learning models mentioned above are currently facing the challenge of **backdoor attack**, wherein attackers deliberately corrupt the models by training them on manipulated datasets. These datasets contain hidden

triggers embedded in input samples from a source class, which cause the network to misclassify these input samples as a different, often target, class. The majority of existing backdoor attack techniques operate in two main ways: first, by retraining a neural network on datasets containing samples with either visible or imperceptible triggers [11], [23] [21]; second, by directly or indirectly modifying the model's weight values through adversarial sampling [8, 23, 34, 39, 41]. The attack methods mentioned earlier have a notable drawback: their backdoor effects can be easily countered by re-training the deep learning models. Furthermore, a model's accuracy is not only affected by its weights but also by its architecture, including the configuration and connections among its layers. To address this problem, [2] introduced an innovative architectural backdoor attack. This method deceives an AlexNet model by injecting high values into the feature map vector following the average pooling layer, triggered by the presence of a hidden trigger in an input image. However, the adversarial images generated by this architectural backdoor attack are visually detectable and cannot be directly used against the VMamba method, as it targets the average pooling layer, which is absent in the VMamba model. Given the impressive performance of the VMamba model compared to ViT models, it is essential to assess its robustness against current backdoor attacks to ensure effective real-time application deployment.

To the best of our knowledge, the VMamba model's resistance to both training-based and architectural backdoor attacks has not been extensively investigated. Consequently, we first evaluated and benchmarked the VMamba model's performance against several prominent backdoor attacks, including BadNets [11], WaNet [23], and R-Fool [21]. Additionally, we introduced a novel architectural backdoor attack that successfully deceives the VMamba model with a higher success rate than existing backdoor attack methods. Our proposed new attack is visually imperceptible, weight-agnostic, and remains effective even after re-training, unlike attacks such as BadNets [11], WaNet [23], and R-Fool [21]. Our proposed backdoor attack significantly enhances stealth compared to current attack methods.

Our proposed method provides several notable contributions: Firstly, we introduced a novel weight-agnostic and imperceptible architectural backdoor attack specifically designed for the visual state space model, with the backdoor seamlessly integrated into the model's architecture. Secondly, we developed four distinct scanning patterns, Viz. Random Efficient Scan (RES), Random Efficient Addition Scan (REAS), Random Efficient Multiply Scan (REMS), and Random Efficient Dropping Scan (REDS), collectively known as "BadScan." These patterns function within the newly designed BadScan block to effectively deceive the VMamba model. Thirdly, we evaluated the robustness of the VMamba model against four backdoor attacks: BadNets, WaNet, R-Fool, and our proposed architectural backdoor attack (BadScan). Additionally, we compared the VMamba model's robustness with that of other deep models, including ViT, CNN-based models, and MLP-mixer, under the same attacks. Finally, we demonstrated that the BadScan-based attack not only survives retraining but also achieves a higher Triggered Accuracy Ratio (TRA) compared to three state-of-the-art backdoor attacks across multiple datasets.

2. Related Work

2.1. SSM in Vision

The exceptional performance of state space models in handling long sequences has led researchers across various fields to leverage these models for establishing new benchmarks. Recent state space models such as Vision Mamba [43], U-Mamba [22], ViViM [3], Medmamba [37], and RS-Mamba [40] have been introduced for applications in image classification, segmentation, video understanding, medical image diagnosis, and remote sensing. A comprehensive review of the latest state space models across different domains is available in [36].

2.2. Backdoor Attacks on VSS Models

Chengbin et al. [5] conducted an empirical study on the robustness of these models against adversarial images, which are generated by introducing imperceptible noise using various techniques, such as FGSM [5], BIM [5], and C&W [5]. Generating adversarial images usually necessitates access to model details such as logits and weights. In contrast, the effectiveness of visual state space models in defending against backdoor attacks has not been thoroughly investigated and requires comprehensive examination. **To address this gap, we assessed the VMamba model's [43] vulnerability to several backdoor attacks [11], [23] [21] and developed a novel architectural backdoor attack that is weight-agnostic, visually imperceptible, and remains effective even after retraining.** Additionally, we have developed four distinct scanning methods that successfully mislead the visual state space model (VMamba), resulting in a higher TRA compared to existing backdoor attacks when applied to VMamba.

2.3. Types of Backdoor Attacks

Backdoor attack techniques can be divided into three categories: attacks with triggers, architectural attacks, and attacks without triggers. In attacks with triggers [6, 26], specific triggers are inserted into the network layers and are activated when the model encounters a particular input. In attacks without triggers, triggers are introduced into the training data as small noise patches or signals [11], [23],

[21]. The model, trained on this poisoned dataset, misclassifies samples into a target class when it detects the trigger. In [2], the authors proposed an architectural backdoor attack in which a trigger, such as a small white or checkerboard patch, is inserted into an image. When this trigger is detected, a predefined function significantly amplifies the value in the feature map after the average pooling layer of the AlexNet model. This method has certain drawbacks: the trigger is visually noticeable, and the attack cannot be executed if the model does not include an average pooling layer, which is the case for VMamba. **In contrast, the proposed attack method employs a visually imperceptible trigger and utilizes a bit-level trigger detection algorithm, making it faster compared to previous methods that rely on statistical techniques and are slower in trigger detection [2]. Additionally, the proposed attack method is compatible with VMamba, overcoming the applicability limitation.**

2.4. Defense Against Backdoor Attacks

A common strategy to protect deep models from adversarial attacks is adversarial training [9], which involves retraining the model on a dataset containing both clean and adversarial samples. However, specific methods to defend against backdoor attacks are still lacking. Pruning-based methods, trigger removal methods, and knowledge distillation-based methods are popular defense techniques for protecting deep models from backdoor attacks. Knowledge distillation (KD)-based techniques involve transferring knowledge from a teacher model trained on clean data to help the backdoored model forget information about the inserted triggers [19, 35]. Trigger removal methods [33, 42] work at the input layer to eliminate triggers embedded in the input, transforming an attacked input into a clean one and preventing the model from displaying abnormal behavior. For example, [12] details a method that monitors consistency across the logits of the deep model to remove triggers from inputs. Pruning-based methods [1, 20] focus on eliminating inactive neurons during the classification of clean inputs, effectively erasing backdoor characteristics and reducing the size of the compromised model. In our work, we first subjected the VMamba model to both the proposed and existing backdoor attacks. Subsequently, we employed attention-blocking [28] and token-dropping [28] defense techniques to protect the VMamba model from different backdoor attacks.

3. Methodology

3.1. State Space Model

The deep networks based on the SSM [17] discussed earlier rely on a conventional continuous system that maps a 1-dimensional input function or sequence, denoted as $x(t) \in \mathbb{R}$, through intermediate latent states $u(t) \in \mathbb{R}$ to an output $y(t) \in \mathbb{R}$. This process is described by a linear Ordinary Differential Equation (ODE):

$$u'(t) = \boldsymbol{E}u(t) + \boldsymbol{F}x(t) \tag{1}$$

$$y(t) = \boldsymbol{G}u(t) \tag{2}$$

Here, $\boldsymbol{E} \in \mathbb{R}^{M\times M}$ represents the state matrix, while $\boldsymbol{F} \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{G} \in \mathbb{R}^{1\times M}$ denote the projection parameters. To adapt this continuous system for deep learning scenarios, S4 and Mamba discretize it. This adaptation involves introducing a timescale parameter $\triangle$ and converting $\boldsymbol{E}$ and $\boldsymbol{F}$ into discrete parameters $\overline{\boldsymbol{E}}$ and $\overline{\boldsymbol{F}}$ using a fixed discretization method. Typically, the zero-order hold (ZOH) method is applied for discretization, defined as follows:

$$\overline{\boldsymbol{E}} = \exp(\triangle \boldsymbol{E}) \tag{3}$$

$$\overline{\boldsymbol{F}} = (\triangle \boldsymbol{E})^{-1}(\exp(\triangle \boldsymbol{F}) - \boldsymbol{I}) \cdot \triangle \boldsymbol{F} \tag{4}$$

The discrete representation of this linear system is given by:

$$u'(t) = \overline{\boldsymbol{E}}u(t) + \overline{\boldsymbol{F}}x(t) \tag{5}$$

$$y(t) = \boldsymbol{G}u(t) \tag{6}$$

$$\boldsymbol{v} = (\boldsymbol{G}\overline{\boldsymbol{F}}, \boldsymbol{G}\overline{\boldsymbol{E}}\overline{\boldsymbol{E}}, \ldots, \boldsymbol{G}\overline{\boldsymbol{E}}^{L-1}\overline{\boldsymbol{F}}) \tag{7}$$

$$y = c_1 \cdot \boldsymbol{v} \tag{8}$$

Here, $\boldsymbol{v} \in \mathbb{R}^M$ represents a structured convolutional kernel, and M denotes the length of the input sequence x.

3.2. BadScan Attack

The BadScan attack method designed to deceive the VMamba model operates in two phases: first, it involves generating and detecting a hidden trigger; second, it disrupts the scan pattern of a VSS block in the VMamba model by substituting the SS2D-based VSS block with the proposed BadScan block. The specifics of trigger detection and the BadScan block are outlined as follows:

3.2.1 Generation, Insertion and Detection of a Trigger

Figure 1 shows the process of creating and detecting triggered patches in an image using the bit plane slicing technique [38]. In our work, the triggers can be inserted at any two predefined locations within an image, which are known only to the attacker. The steps involved in the creation and detection of a hidden trigger in an image are as follows:

1. Let $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ denote the two patches chosen from two predefined locations in an image. The bit planes $\boldsymbol{B}$ for these patches, obtained using the bit plane slicing

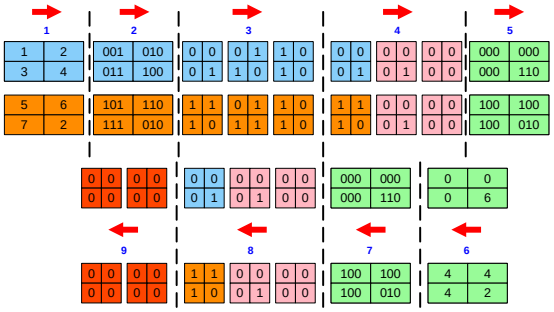


Figure 1. Workflow of Trigger Generation and Detection (Illustrated with a 3-Bit Representation for Better Clarity)

function (f_S), can be represented as follows **(Steps 1, 2 and 3 of Figure 1):**

$$\boldsymbol{B}_7^i, \boldsymbol{B}_6^i, ..., \boldsymbol{B}_1^i, \boldsymbol{B}_0^i = f_S(\boldsymbol{P_i}) \tag{9}$$

$$\boldsymbol{B}_7^j, \boldsymbol{B}_6^j, ..., \boldsymbol{B}_1^j, \boldsymbol{B}_0^j = f_S(\boldsymbol{P_j}) \tag{10}$$

Here, the leftmost bit planes ($\boldsymbol{B}_7^j$) are the most significant bit planes (MSBPs), while the rightmost bit planes ($\boldsymbol{B}_1^j$) are the least significant bit planes (LSBPs). In this work, each pixel value is represented using eight bits, resulting in a total of eight-bit planes for each selected patch.

2. In the next step, a bitwise XOR operation is performed between the k^{th} bit planes of patches $\boldsymbol{P_i}$ and $\boldsymbol{P_j}$. This process is outlined as follows **(Step 4 of Figure 1):**

$$\overline{\boldsymbol{B}}_k^i = \overline{\boldsymbol{B}}_k^j = \boldsymbol{B}_k^i \oplus \boldsymbol{B}_k^j \tag{11}$$

3. The next step involves generating the triggered patches $\boldsymbol{P}_i^{'}$ and $\boldsymbol{P}_j^{'}$ by replacing the k^{th} bit plane $\boldsymbol{B}_k^i$ of patch $\boldsymbol{P}_i$ with its modified k^{th} bit plane, $\overline{\boldsymbol{B}}_k^i$. A similar modification is applied to the k^{th} bit plane $\boldsymbol{B}_k^j$ of patch $\boldsymbol{P}_j$. Consequently, the modified triggered patches $\boldsymbol{P}_i^{'}$ and $\boldsymbol{P}_j^{'}$ can be represented as follows **(Step 5 of Figure 1):**

$$\boldsymbol{P}_i^{'} = f_S^{-1}(\boldsymbol{B}_7^i, \boldsymbol{B}_6^i, ..., \overline{\boldsymbol{B}}_1^i, \overline{\boldsymbol{B}}_0^i) \tag{12}$$

$$\boldsymbol{P}_j^{'} = f_S^{-1}(\boldsymbol{B}_7^j, \boldsymbol{B}_6^j, ..., \overline{\boldsymbol{B}}_1^j, \overline{\boldsymbol{B}}_0^j) \tag{13}$$

Here f_S^{-1} indicates the reverse operation of the bit plane slicing.

4. The modified patches ($\boldsymbol{P}_i^{'}$ and $\boldsymbol{P}_j^{'}$) are placed into the predefined locations of the original patches ($\boldsymbol{P}_i$ and $\boldsymbol{P}_j$) to generate a visually imperceptible attacked image from the clean image.

5. To detect the trigger, the k^{th} bit planes of the i^{th} and j^{th} patches from predefined locations in the input image are selected, and a bitwise XOR operation is performed between them to generate the k^{th} trigger matrix $\boldsymbol{T}$. If the resulting trigger matrix is a zero matrix, the trigger is detected; otherwise, the trigger is absent from the input image. The trigger matrix $\boldsymbol{T}_k$ is computed using the following expression **(Steps 6, 7, 8 and 9 of Figure 1):**

$$\boldsymbol{T}_k = \overline{\boldsymbol{B}}_k^i \oplus \overline{\boldsymbol{B}}_k^j \tag{14}$$

3.2.2 BadScan Block

The main idea behind the proposed architectural backdoor attack is to interfere with the scanning pattern of the visual state space (VSS) block [43] in the VMamba model upon detection of a trigger within the input image. Figure

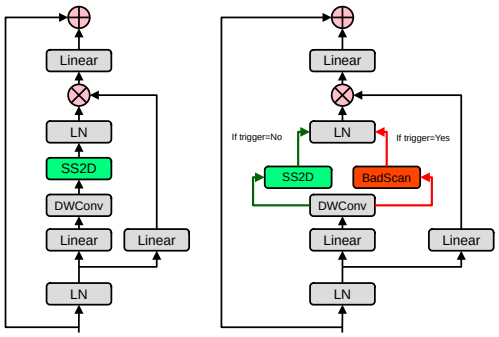


Figure 2. SS2D-Based VSS Block (Left) [43] Vs BadScan Block (Right)

2 illustrates the difference between the standard VSS block and the VSS block affected by the proposed attack. In the absence of an imperceptible hidden trigger, patch scanning proceeds using the standard SS2D-based VSS block (indicated in green). However, when an imperceptible hidden trigger is present, the proposed BadScan block (indicated in red) takes over the scanning process. Figure 3 illustrates the working of the four proposed scanning patterns within the BadScan block for a given set of image patches. The details of the four proposed scanning patterns are as follows:

1. **Random Efficient Scan (RES)**: This scanning method employs an efficient scan [24] as its base technique. The key distinction is that, in this approach, the patches corresponding to the four output scan patterns of the efficient scan are selected randomly, with repetition of patches permitted.

2. **Random Efficient Addition Scan (REAS)**: This scanning method is similar to the RES scan, with the main difference being that each patch corresponding to

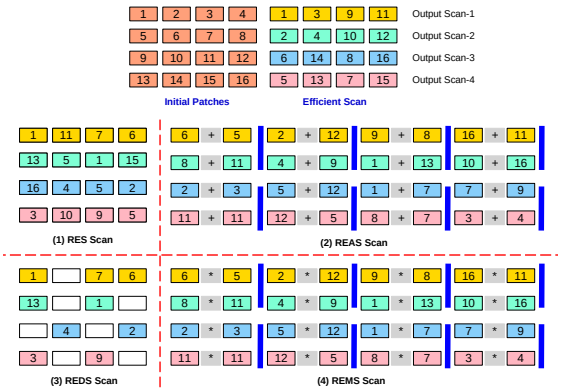


Figure 3. Working of RES, REAS, REMS and REDS Scanning Patterns Inside BadScan Block

the four output scan patterns is the result of adding any two randomly selected patches.

3. **Random Efficient Patch Dropping Scan (REDS)**: This scanning method is similar to the RES scan but with a key difference: initially, each of the patches corresponding to the four output scan patterns is selected randomly. Later, $r\%$ of the patches in each of the four output scan patterns are dropped. In our work the value of r is set to 20%.

4. **Random Efficient Multiply Scan (REMS)**: This scanning method is similar to the RES scan, with the key difference being that each patch corresponding to the four output scan patterns is generated by elementwise multiplication of any two randomly selected patches.

4. Experiments & Results

This section will detail the experimental setup and discuss the results obtained.

4.1. Image Datasets

We selected two widely used image classification datasets, CIFAR-10 [18] and ImageNet-1K [27], to evaluate the performance of the VMamba model against both existing and proposed backdoor attacks. The poison ratio is set at 0.33 when assessing VMamba's robustness against BadNets [11], WaNet [23], R-Fool [21] attacks. All images are resized and transformed to a dimension of 224×224. We used the same set of source-target pairs as those in [11] to ensure a fair comparison. The performance results for the BadNet, R-Fool, and WaNet attacks are averaged across the three sets of source-target pairs for both CIFAR-10 [18] and ImageNet-1K [27]. The source-target pairs selected for each dataset are as follows: For CIFAR-10, the pairs are **(Cat, Truck), (Car, Dog), and (Deer, Ship)**. In the case of the ImageNet-1K dataset, the chosen pairs are **(Seat Belt, Computer), (Shih-Tzu, Greyhound Racing), and (Bell Pepper, Chess Master)**.

4.2. Deep Networks

In addition to evaluating the VMamba model's [43] robustness against the four backdoor attacks, we also compared its robustness with that of ResNet-50 [13], ResNet-18 [13], ViT [4], and MLP-mixer [31] architectures.

4.3. Experimental Details

In our work, we utilized a system with 128GB of RAM, running Ubuntu 22.04 LTS, and equipped with an NVIDIA GeForce RTX 4090 GPU. All models were trained using the SGD optimizer with a momentum of 0.90 and a learning rate 0.001. The scripts were written in PyTorch 2.2.0, and the attacked models were trained for ten epochs. For the BadNets attack, the patch size was set to 30, with other hyperparameters following those specified in [11]. For the proposed BadScan attack, the patch size was configured to (4×4) across all three channels. The patches were placed in the top-left and bottom-left corners of the image.

4.4. Evaluation

The robustness of the VMamba and other deep models is assessed using the three metrics [2], viz. **Clean Task Accuracy (CTA)**, **Triggered Task Accuracy (TTA)** and **Triggered Accuracy Ratio (TAR)**. The CTA is defined as the accuracy of a model when evaluated on clean test data samples. The TTA is defined as the accuracy of a model when evaluated on test data samples that contain triggers. The TAR is defined as the ratio of CTA to TTA. It indicates the relative decrease in performance that a backdoor attack causes when a deep model is fed input samples with triggers. A higher CTA combined with a lower TTA signifies an ideal backdoor attack that effectively deceives a deep model. Similarly, a high TAR value also indicates a highly effective backdoor attack.

4.5. VMamba Under Backdoor Attacks

Table 1 illustrates the performance of various deep models against different backdoor attacks. For the VMamba model, the TAR values for BadNets, WaNet, R-Fool, and BadScan are 1.33, 1.14, 1.20, and 15.50, respectively, for the ImageNet-1K dataset. For the same attacks, the TAR values for the CIFAR-10 dataset are 1.40, 1.07, 1.87, and 6.27, respectively. It is evident that the VMamba model consistently achieves the lowest TAR values for BadNets, WaNet, and R-Fool attacks, indicating that it is more robust against these backdoor attacks compared to the other five deep models. Among the patch-based models, Viz. MLP-mixer, ViT, and V-Mamba, the robustness, measured

Table 1. Performance of Deep Models Against Backdoor Attacks

Attacks	Model	ImageNet-1K			CIFAR-10		
		CTA	TTA	TAR	CTA	TTA	TAR
BadNets [11]	ResNet-18	72.03	46.27	1.56	85.43	46.77	1.83
	ResNet-50	76.00	47.80	1.59	83.97	15.13	5.55
	MLP-mixer	70.40	50.83	1.38	95.17	7.00	13.60
	ViT-S	70.37	46.67	1.51	94.40	50.33	1.88
	VMamba	66.13	49.80	**1.33** ↑ **(14.17)**	93.40	66.53	**1.40** ↑ **(4.87)**
WaNet [23]	ResNet-18	72.13	54.07	1.33	84.47	52.23	1.61
	ResNet-50	75.20	53.53	1.40	84.57	17.53	4.82
	MLP-mixer	66.67	38.93	1.71	91.93	71.37	1.29
	ViT-S	73.17	60.20	1.22	94.43	74.00	1.28
	VMamba	67.38	59.20	**1.14** ↑ **(14.39)**	93.47	86.97	**1.07** ↑ **(5.20)**
Refool [21]	ResNet-18	72.87	41.37	1.76	87.73	35.23	2.49
	ResNet-50	75.40	48.97	1.54	83.73	8.37	10.01
	MLP-mixer	69.10	38.90	1.78	94.30	32.57	2.90
	ViT-S	72.63	49.47	1.47	96.47	38.20	2.53
	VMamba	65.73	54.80	**1.20** ↑ **(14.30)**	94.93	50.73	**1.87** ↑ **(4.40)**
BadScan	VMamba	93.00	6.00	**15.50**	94.00	15.00	**6.27**
	MiM	90.00	6.00	**15.00**	94.00	8.00	**11.75**
	EF-Mamba	90.00	5.00	**18.00**	98.00	11.00	**8.91**

by the TAR, is ranked in decreasing order as follows: V-Mamba, ViT, and MLP-mixer. This order remains consistent for BadNets, WaNet, and R-Fool attacks across both datasets. We also assessed the BadScan method's performance relative to other variants of the VMamba model, including Efficient-VMamba (EF-Mamba) [24] and Mamba-in-Mamba (MiM) [3]. The BadScan method effectively deceives both the EF-Mamba and MiM models across the datasets, achieving a TAR comparable to that of the VMamba model.

4.6. Ablation Study

4.6.1 Effectiveness in Other Domains

In addition to assessing the performance of the proposed BadScan method in computer vision tasks, we also evaluated its effectiveness in other domains, such as audio classification. For this purpose, we used the EPIC-Sound Dataset [16] and the Audio Mamba [7] model. The EPIC-Sound dataset includes audio recordings of various kitchen activities, such as washing and opening the fridge. For the EPIC-Sound dataset [16], the selected pairs include **(Cut Chop, Rustle), (Metal Collision, Drawer Open), and (Scrub, Tap Water)**. Table 2 displays the performance of the Audio Mamba model when subjected to the BadScan attack. The REDS scan-based BadScan method achieved the best results. This demonstrates that BadScan effectively deceives the Audio Mamba model, highlighting its potential applicability across different domains.

4.6.2 Impact of Number of Selected Bit Planes

Tables 3, 4, and 5 examine how the number of bit planes selected during trigger insertion affects the performance of the BadScan attack on the VMamba, MiM, and EF-Mamba models, respectively. Based on the TAR values for different k values across both datasets and the VMamba model and its variants, the following observations can be made. For the VMamba and EF-Mamba models, the REDS scan-based BadScan outperformed the other three scan-based BadScan attacks in deceiving the VMamba and EF-Mamba models across both datasets. In the case of the MiM model, the best performance of the BadScan attack was achieved with the REDS scan for ImageNet-1K and the REAS scan for CIFAR-10. It is worth noting that, as evident from the tables mentioned above, regardless of the number of bit planes or scan type, the BadScan attack effectively deceives the VMamba and its variants across both datasets, achieving higher TAR values compared to BadNets, WaNets, and R-Fool attacks.

Table 2. BadScan Attack on Audio-Mamba For Different (k)

Dataset	BadScan Type	CTA				TTA				TAR			
		K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
EPIC-Sound	RES	82.00	86.00	81.00	83.00	19.00	20.00	15.00	11.00	4.32	4.30	5.40	7.55
	REAS	83.00	78.00	84.00	85.00	20.00	18.00	12.00	20.00	4.15	4.33	7.00	4.25
	REMS	82.00	75.00	84.00	84.00	15.00	20.00	20.00	23.00	5.47	3.75	4.20	3.65
	REDS	78.00	81.00	79.00	84.00	10.00	15.00	12.00	20.00	7.80	5.40	6.58	4.20

Table 3. BadScan Attack on VMamba For Different (k)

Dataset	BadScan Type	CTA				TTA				TAR			
		K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
ImageNet-1K	RES	96.00	89.00	85.00	88.00	15.00	5.00	9.00	6.00	6.4	17.80	9.44	14.67
	REAS	94.00	90.00	89.00	88.00	12.00	16.00	8.00	10.00	7.83	5.63	11.13	14.67
	REMS	92.00	91.00	93.00	87.00	15.00	7.00	12.00	6.00	6.13	13.00	7.75	14.50
	REDS	90.00	91.00	93.00	91.00	7.00	8.00	6.00	6.00	12.86	11.38	15.50	15.17
CIFAR-10	RES	81.00	92.00	88.00	87.00	29.00	25.00	26.00	17.00	2.79	3.68	3.38	5.12
	REAS	83.00	89.00	91.00	92.00	17.00	15.00	20.00	13.00	4.88	5.93	4.55	7.08
	REMS	77.00	85.00	85.00	84.00	19.00	16.00	24.00	18.00	4.05	6.27	3.54	4.67
	REDS	89.00	94.00	91.00	89.00	15.00	15.00	15.00	15.00	5.93	6.27	6.07	5.93

Table 4. BadScan Attack on MiM For Different (k)

Dataset	BadScan Type	CTA				TTA				TAR			
		K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
ImageNet-1K	RES	92.00	89.00	87.00	87.00	10.00	5.00	8.00	8.00	9.20	17.80	10.88	10.88
	REAS	92.00	90.00	90.00	89.00	12.00	12.00	12.00	15.00	7.67	7.50	7.50	5.93
	REMS	91.00	88.00	90.00	90.00	11.00	6.00	15.00	8.00	8.27	14.67	6.00	11.25
	REDS	90.00	90.00	87.00	95.00	6.00	7.00	8.00	10.00	15.00	12.86	10.88	9.50
CIFAR-10	RES	90.00	96.00	91.00	93.00	8.00	13.00	20.00	12.00	11.25	7.38	4.55	7.75
	REAS	94.00	95.00	93.00	89.00	8.00	11.00	9.00	11.00	11.75	8.64	10.33	8.09
	REMS	89.00	90.00	90.00	86.00	15.00	22.00	20.00	16.00	5.93	4.09	4.50	5.38
	REDS	88.00	93.00	89.00	88.00	16.00	11.00	15.00	14.00	5.50	8.45	5.93	6.29

Table 5. BadScan Attack on EF-Mamba For Different (k)

Dataset	BadScan Type	CTA				TTA				TAR			
		K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
ImageNet-1K	RES	92.00	93.00	90.00	94.00	14.00	10.00	15.00	6.00	6.57	9.30	6.00	14.67
	REAS	97.00	96.00	95.00	89.00	15.00	10.00	15.00	13.00	6.47	9.60	6.33	6.85
	REMS	92.00	92.00	96.00	90.00	12.00	9.00	10.00	6.00	7.67	10.22	9.60	15.00
	REDS	90.00	96.00	92.00	94.00	5.00	7.00	6.00	10.00	18.00	13.71	15.33	9.40
CIFAR-10	RES	78.00	85.00	84.00	88.00	28.00	31.00	19.00	23.00	2.79	2.74	4.42	3.83
	REAS	83.00	90.00	90.00	77.00	19.00	26.00	13.00	17.00	4.37	3.46	6.92	5.41
	REMS	78.00	81.00	77.00	79.00	20.00	20.00	30.00	15.00	3.90	2.57	2.57	5.27
	REDS	94.00	98.00	89.00	90.00	15.00	11.00	20.00	13.00	6.27	8.91	4.45	6.92

4.7. Performance of Defense Methods

Table 6 illustrates the effectiveness of attention-blocking [28] and token-dropping defense [28] methods against three different backdoor attacks when applied to the ViT and VMamba models on the ImageNet-1K dataset. It is evident from Table 6 that the TTA values increased significantly for both models across all three attacks. However, despite their effectiveness, these methods rely on the weight information of the backdoored model to secure the deep model, making them inadequate for defending against the proposed BadScan attack, which operates at the architectural level during test time. To address this, a potential area for future research could be the development of a defense mechanism specifically aimed at countering the BadScan attack.

Table 6. Performance of Different Defense Methods

Models	ViT			VMamba		
Attacks	BadNets	WaNet	R-Fool	BadNets	WaNet	R-Fool
Defense Method	TTA	TTA	TTA	TTA	TTA	TTA
Attention Blocking	55.00	92.33	65.67	76.33	60.00	87.04
Token Dropping	46.00	91.00	66.00	74.20	86.00	58.70
No Defense	46.67	60.20	49.47	49.80	59.20	54.80

4.8. Attack Crafting Time

The attack crafting time required by BadNets, WaNet, and R-Fool are $5.72e^{-6}$, 0.5325 and 0.0119 seconds, respectively. For the BadScan attack, the times required for k values of $1, 3, 5$, and 7 are $0.000427, 0.000439, 0.000438$, and 0.000449 seconds, respectively. Among these methods, BadNets is the fastest, while WaNet is the slowest. Notably, the proposed BadScan attack is the second fastest after BadNets and achieves the highest TAR in deceiving the VMamba model. Additionally, the time required for BadScan increases with the number of bit planes (k) used to craft the backdoored sample. The reason for proposing our own trigger detection algorithm is that our detection method requires only 0.00036 seconds (approximately three times faster than the detection method described in [2]), whereas the trigger detector in [2] takes 0.0122 seconds. Additionally, unlike our trigger, which is visually imperceptible, the trigger in [2] is visually detectable.

4.9. PSNR Values

Maintaining a high PSNR value is essential for preserving the stealth of an attack. For ImageNet-1K, the PSNR values [14] for the BadNets, WANet, R-Fool, and BadScan (REDS, $k = 1$) attacks are 40.79, 38.09, 27.94, and 43.10, respectively. For CIFAR-10, the PSNR values for these same attacks are 43.75, 45.77, 28.50, and 49.11, respectively. Among the various attack methods, BadScan consistently achieves the highest PSNR values for both datasets, indicating that the hidden trigger introduced by BadScan remains visually imperceptible in the backdoored images.

4.10. Persistence Against Retraining

We also evaluated the performance of the VMamba model under two conditions. First, we analyzed the model, initially trained on ImageNet-1K, after it was attacked with BadNet (using ImageNet-1K images) and then fine-tuned on the CIFAR-10 dataset. Second, we assessed the performance of the same attacked VMamba model after it was retrained from scratch on the CIFAR-10 dataset. Table 7 presents a comparison of four attacks based on their effectiveness following the retraining of the VMamba model. The CTA and TTA values for the VMamba model remained comparable in both scenarios, indicating that the backdoor effects of BadNets, WaNet, and R-Fool were fully mitigated during either retraining from scratch or fine-tuning, as the backdoor-related weights were entirely removed. In contrast, the BadScan attack remained effective, significantly reducing the TTA value and achieving the highest TAR values of 9.79 and 6.27 for the first and second settings, respectively. This indicates that the proposed backdoor attack continues to be effective even after the model has been retrained or fine-tuned.

Table 7. Impact of Retraining and Fine-Tuning

Attack	Fine-Tunning			From Scratch		
	CTA	TTA	TAR	CTA	TTA	TAR
BadNets	80.10	78.40	1.02 ↑ (8.77)	93.80	93.60	1.01 ↑ (5.26)
WaNet	80.50	82.80	0.98 ↑ (8.81)	93.40	93.80	0.99 ↑ (5.28)
R-Fool	81.20	67.80	1.19 ↑ (8.60)	94.00	93.80	0.99 ↑ (5.28)
BadScan	91.10	9.26	**9.79**	94.00	15.00	**6.27**

4.11. Qualitative Analysis

The primary goal of a backdoor attack is to establish a strong association between an image and a concealed trigger, ensuring that the attacked model consistently classifies a source image into a target class whenever the hidden trigger is detected. For the CIFAR-10 and ImageNet-1K datasets, the source class is set to **Deer** and **Shih Tzu**, respectively. The target classes for these datasets are **Ships** and **Greyhound Racing**. Figures 4 and 6 illustrate the clean images and those with backdoor attacks for the CIFAR-10 and ImageNet-1K datasets, respectively. From these figures, it is evident that the hidden triggers inserted by Badnets, WaNets, and R-Fool are visually noticeable. In contrast, the hidden trigger crafted by BadScan is visually imperceptible across the backdoored images which is more preferable. The effectiveness of an inserted trigger in fooling a model is demonstrated when the model starts focusing on the region where the trigger is present in a backdoored image. This behavior can be validated using Gradient-weighted Class Activation Mapping (Grad-CAM)

(a) Clean (b) BadNets (c) WaNet (d) R-Fool (e) BadScan (REDS)

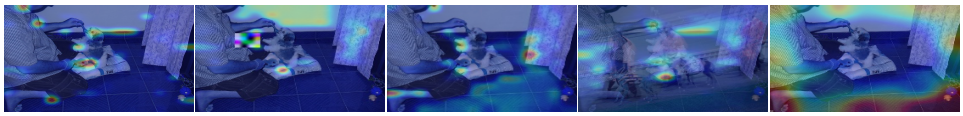
Figure 4. Clean and Attacked Images from CIFAR-10 (Target Class= Ships, Source Class= Deer)

(a) Clean (b) BadNets (c) WaNet (d) R-Fool (e) BadScan (REDS)

Figure 5. Grad-CAM Maps of Clean and Attacked Images from CIFAR-10

(a) Clean (b) BadNets (c) WaNet (d) R-Fool (e) BadScan (REDS)

Figure 6. Clean and Attacked Images from ImageNet-1K (Target Class= Greyhound Racing, Source Class=Shih Tzu)



(a) Clean (b) BadNets (c) WaNet (d) R-Fool (e) BadScan (REDS)

Figure 7. Grad-CAM Maps of Clean and Attacked Images from ImageNet-1K

plots, which illustrate the areas of focus for both clean and backdoored images. Figure 5 and 7 illustrate the Grad-CAM analysis of a clean and its backdoored image for the CIFAR-10 and its ImageNet-1K datasets, respectively for the VMamba network. Figure 5(a) illustrates that the model primarily focuses on the body region of a deer. In contrast, Figures 5(b), 5(c), and 5(d) show that the model focuses more on the regions with hidden triggers when subjected to the BadNets, WaNet, and R-Fool attacks. For the BadScan attack (Figure 5(e)), however, the model predominantly focuses on regions outside the deer's body, rather than on the regions where hidden triggers are present in the other attacks. Similar behavior is observed for the ImageNet-1K with the VMamba model across Figures 7(a), 7(b), 7(c) and 7(d), corresponding to the scenarios with no attack, BadNet, WaNet, and R-Fool attacks, respectively. In the absence of an attack, the VMamba focuses more on the region with the Shih Tzu. However, with the BadNet, WaNet, and R-Fool attacks, the VMamba shifts its focus to regions containing the hidden trigger. For the BadScan attack (Figure 7(e)), the VMamba again highlights regions outside the Shih Tzu's body, rather than focusing on the regions where hidden triggers are present in the other attacks. In a successful backdoor attack, the model's attention may shift to specific regions associated with the backdoor trigger, rather than focusing on the actual features of the source class. Grad-CAM visualizations can reveal this shift by showing that the model's attention is redirected to features characteristic of the target class (e.g., focusing on areas resembling parts of a Ship instead of a Deer). These observations are reinforced by the Grad-CAM visualization of the VMamba model under the BadScan attack. Further analysis and results are provided in the supplementary material of this work.

5. Conclusion

In this paper, we introduce BadScan, a novel architectural backdoor attack aimed at deceiving the visual state space model. The proposed method utilizes bit plane slicing to embed a visually imperceptible hidden trigger within an image. A similar approach is used to detect the trigger within an input image, which then activates the BadScan attack if the trigger is found. The BadScan is weight-agnostic and retains its effectiveness even after the model undergoes retraining. Our experiments and results on three different datasets reveal two key findings: First, the VMamba model shows considerable robustness against existing backdoor attacks. Second, the proposed BadScan attack outperforms current backdoor attacks and effectively misleads the visual state space model with a high triggered accuracy ratio, thereby presenting a significant threat to the visual state space model and its variants. Furthermore, the BadScan attack effectively deceives the visual state space model even when applied to datasets from other domains, such as audio classification. The proposed BadScan attack has two limitations. First,

it is currently untargeted, meaning it does not target specific classes or outcomes. Second, the attacker must be aware of the locations of the triggered patches within the images to execute the attack effectively. We hope that our proposed attack will inspire the vision community to create robust defense mechanisms, such as neural architecture search-based defenses or the designing of weight-agnostic networks, to secure visual state space models from BadScan and other advanced backdoor attack methods.

References

[1] Peter Bajcsy and Michael Majurski. Baseline pruning-based approach to trojan detection in neural networks. *ArXiv*, abs/2101.12016, 2021. 3

[2] Mikel Bober-Irizar, Ilia Shumailov, Yiren Zhao, Robert Mullins, and Nicolas Papernot. Architectural backdoors in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24595–24604, 2023. 2, 3, 5, 7

[3] Tianxiang Chen, Zhentao Tan, Tao Gong, Qi Chu, Yue Wu, Bin Liu, Jieping Ye, and Nenghai Yu. Mim-istd: Mamba-in-mamba for efficient infrared small target detection. *arXiv preprint arXiv:2403.02148*, 2024. 1, 2, 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 5

[5] Chengbin Du, Yanxi Li, and Chang Xu. Understanding robustness of visual state space models for image classification, 2024. 2

[6] Jacob Dumford and Walter Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2020. 2

[7] Mehmet Hamza Erol, Arda Senocak, Jiu Feng, and Joon Son Chung. Audio mamba: Bidirectional state space model for audio representation learning. *arXiv preprint arXiv:2406.03344*, 2024. 6

[8] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. Fiba: Frequency-injection based backdoor attack in medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20876–20885, 2022. 2

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 3

[10] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1

[11] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 2, 5, 6

[12] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *The Eleventh International Conference on Learning Representations*. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[14] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 7

[15] Chengxi Huang, Wei Wang, Xin Zhang, Shui-Hua Wang, and Yu-Dong Zhang. Tuberculosis diagnosis using deep transferred efficientnet. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):2639–2646, 2022. 1

[16] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023. 6

[17] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 1, 3

[18] Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, et al. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5):2, 2014. 5

[19] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021. 3

[20] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018. 3

[21] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 182–199. Springer, 2020. 2, 3, 5, 6

[22] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024. 1, 2

[23] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021. 2, 5, 6

[24] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024. 4, 6

[25] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 367–376, 2021. 1

[26] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[28] Akshayvarun Subramanya, Soroush Abbasi Koohpayegani, Aniruddha Saha, Ajinkya Tejankar, and Hamed Pirsiavash. A closer look at robustness of vision transformers to backdoor attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3874–3883, 2024. 3, 7

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1

[30] Rui Tian, Zuxuan Wu, Qi Dai, Han Hu, Yu Qiao, and Yu-Gang Jiang. Resformer: Scaling vits with multi-resolution training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22721–22731, 2023. 1

[31] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 1, 5

[32] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. 1

[33] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. 3

[34] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision*, pages 396–413. Springer, 2022. 2

[35] Jun Xia, Ting Wang, Jiepin Ding, Xian Wei, and Mingsong Chen. Eliminating backdoor triggers for deep neural networks using attention relation graph distillation. *arXiv preprint arXiv:2204.09975*, 2022. 3

[36] Rui Xu, Shu Yang, Yihui Wang, Bo Du, and Hao Chen. A survey on vision mamba: Models, applications and challenges. *arXiv preprint arXiv:2404.18861*, 2024. 2

[37] Yubiao Yue and Zhenzhang Li. Medmamba: Vision mamba for medical image classification. *arXiv preprint arXiv:2403.03849*, 2024. 1, 2

[38] Zhe Zhang, Huairui Wang, Zhenzhong Chen, and Shan Liu. Learned lossless image compression based on bit plane slicing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27579–27588, 2024. 3

[39] Feng Zhao, Li Zhou, Qi Zhong, Rushi Lan, and Leo Yu Zhang. Natural backdoor attacks on deep neural networks via raindrops. *Security and Communication Networks*, 2022(1):4593002, 2022. 2

[40] Sijie Zhao, Hao Chen, Xueliang Zhang, Pengfeng Xiao, Lei Bai, and Wanli Ouyang. Rs-mamba for large remote sensing image dense prediction. *arXiv preprint arXiv:2404.02668*, 2024. 1, 2

[41] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 97–108, 2020. 2

[42] Qi Zhou, Zipeng Ye, Yubo Tang, Wenjian Luo, Yuhui Shi, and Yan Jia. Evolutionary trigger detection and lightweight model repair based backdoor defense. 2024. 3

[43] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 1, 2, 4, 5