electronics

MDPI

Article

MalDBA: Detection for Query-Based Malware Black-Box Adversarial Attacks

Zixiao Kong [1], Jingfeng Xue [1], Zhenyan Liu [1,*], Yong Wang [1] and Weijie Han [2]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2] School of Space Information, Space Engineering University, Beijing 101416, China
[*] Correspondence: zhenyanliu@bit.edu.cn

Abstract: The increasing popularity of Industry 4.0 has led to more and more security risks, and malware adversarial attacks emerge in an endless stream, posing great challenges to user data security and privacy protection. In this paper, we investigate the stateful detection method for artificial intelligence deep learning-based malware black-box attacks, i.e., determining the presence of adversarial attacks rather than detecting whether the input samples are malicious or not. To this end, we propose the MalDBA method for experiments on the VirusShare dataset. We find that query-based black-box attacks produce a series of highly similar historical query results (also known as intermediate samples). By comparing the similarity among these intermediate samples and the trend of prediction scores returned by the detector, we can detect the presence of adversarial samples in indexed samples and thus determine whether an adversarial attack has occurred, and then protect user data security and privacy. The experimental results show that the attack detection rate can reach 100%. Compared to similar studies, our method does not require heavy feature extraction tasks or image conversion and can be operated on complete PE files without requiring a strong hardware platform.

Keywords: stateful detection; adversarial defence; artificial intelligence security; privacy protection

check for updates

Citation: Kong, Z.; Xue, J.; Liu, Z.; Wang, Y.; Han, W. MalDBA: Detection for Query-Based Malware Black-Box Adversarial Attacks. *Electronics* **2023**, *12*, 1751. https://doi.org/10.3390/electronics12071751

Academic Editor: Aryya Gangopadhyay

Received: 27 February 2023
Revised: 26 March 2023
Accepted: 4 April 2023
Published: 6 April 2023

1. Introduction

With the advent of the Industry 4.0 era, security threats have increased dramatically, and the number of malware introduced by attackers is rising every year. The volume of malware threats observed by McAfee Labs averaged 688 threats per minute, an increase of 40 threats per minute (3%) in the first quarter of 2021 [1]. VirusTotal's database had more than one million signed samples that were considered suspicious (with more than 15% anti-viruses detecting them as malicious) from January 2021 to April 2022 [2]. Researchers are constantly looking for effective malware detection and classification methods, and with the popularity of artificial intelligence (AI), they find that deep learning-based malware detection and classification methods work well [3–5]. However, deep learning (DL) models are highly vulnerable to adversarial examples [6,7]. Therefore, analyzing and detecting DL-based malware black-box adversarial attacks is a difficult task for anti-malware researchers. The existing optimal defense methods are stateless detection methods such as adversarial retraining and distillation, which detect whether the input sample is benign or malicious without judging whether there is an adversarial attack [8,9]. Existing malware stateful detection methods are implemented in the feature space, which requires data preprocessing and feature extraction [10,11]. At present, there is no malware stateful detection strategy implemented in the problem space.

Driven by this, we propose the MalDBA(Detection for Query-based Malware Black-box adversarial Attacks) to defend against malware black-box adversarial attacks. The process of MalDBA is as follows: First, malicious datasets are obtained from the VirusShare website [12], benign datasets are collected through crawling, and a malware detection model MalConv is pretrained [13]. Then, two different black-box adversarial attacks are reconstructed [14,15], and the history of query results (also known as the intermediate samples) of these attacks are saved. We can find that the prediction scores of these intermediate samples under MalConv model detection are gradually decreasing (meaning that the original malware tends to become a benign-looking sample after adding perturbations). After that, the similarities of the sample sets saved in the query process are compared using the similarity comparator [16]. We find that these intermediate samples are highly similar to each other and the original malicious file, but not similar to other samples. Thus, we can perform the stateful detection of query-based malware black-box adversarial attacks. When it is found that the samples input to the detector model for querying are similar and the predicted scores returned by these similar samples gradually decrease (from malicious to benign), it is judged that the detector is experiencing adversarial attacks.

We evaluated MalDBA on the downloaded dataset and achieved satisfactory results. In summary, the main contributions in this paper are as follows:

(1) We propose MalDBA to defend against query-based malware black-box attacks, which can help analysts effectively detect the existence of adversarial attacks.

(2) We propose a stateful detection method for black-box adversarial attacks. Most of the previous detection methods for adversarial examples (AEs) are stateless, and the method proposed by us can precisely carry out a supplementary defense. The existing stateful detection methods of malware black-box attacks are based on the feature space level, while our method is based on the complete malicious file (i.e., problem space).

(3) We propose a novel similarity comparator based on the MinHash algorithm to analyze the history of queries (i.e., intermediate samples) received by the malware detector.

(4) MalDBA can be run on ordinary personal workstations and does not require high-performance hardware resources, so it meets the needs of ordinary researchers to deal with a large number of malicious codes.

The structure of the article is as follows: In Section 2, we first introduce the necessary background knowledge and the summary of the related work. Section 3 describes the overall framework of the MalDBA. The experimental details are presented in Section 4. Then evaluate it in Section 5. Section 6 discusses some issues. Finally, we conclude in Section 7.

2. Background and Related Work

The adversarial attack and defense of malware is an iterative and complementary process. In recent years, the research of malware black-box attack and detection has emerged [17–19]. To better introduce the content of this paper, we first outline the research background and related work.

2.1. Background

2.1.1. Query-Based Black-Box Attack

Currently, the black-box adversarial attack can be divided into transfer-based attacks and query-based attacks. Transfer-based attacks generate adversarial examples on local surrogate models and directly use the generated adversarial examples to attack the black-box model. However, the attack performance of transfer-based attacks is usually unsatisfactory due to overfitting the local surrogate models. Query-based attacks approximate the gradient information by queries to the target model to craft adversarial examples. Query-based black-box attack is generally divided into decision-based black-box attack and score-based black-box attacks [20]. The decision-based black-box attack, also known as hard-label black-box adversarial attack, iteratively perturbs the original sample by estimating the gradient or boundary proximity and generating AEs according to some strategies [21]. The score-based black-box attack estimates the gradient of the target model loss function according to the output of the target model for the input samples (i.e., the probability scores of each category), and generates the corresponding adversarial samples [22]. Query-based black-box attack often requires multiple queries to generate a successful AE to achieve optimal attack performance. In this paper, the attack we use is a score-based black-box attack, and the attack scenario is shown in Figure 1.

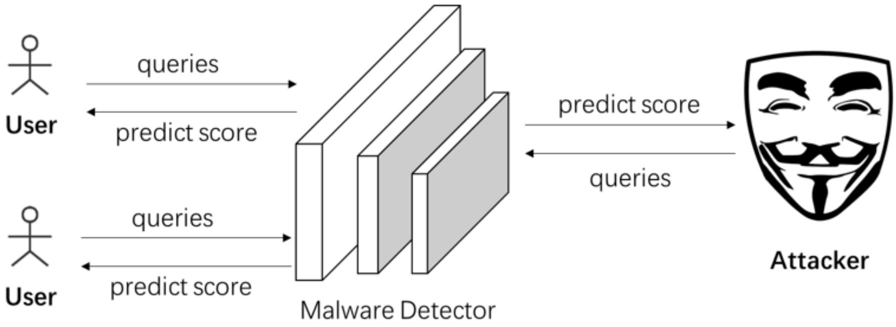


Figure 1. Scenario for score-based malware black-box attacks.

2.1.2. Stateful Detection Method

Stateful detection examines a series of queries submitted by each user to decide whether the user is an attacker [10,23,24]. Given user A and the set of queries he submits, stateful detection checks if an adversarial attack occurred in these queries. Specifically, stateful detection calculates the similarity between queries $q_1, q_2, ..., q_n$ from A. If the similarity exceeds a threshold and the prediction scores returned by the malware detector range from malicious to benign, stateful detection marks A as an adversarial attacker. To calculate the similarity between samples, a similarity comparator is proposed for comparison. In general, the stateful detection method judges whether an adversarial attack has occurred, rather than detecting whether the input samples are malicious.

2.2. Related Work

Research on the detection of adversarial attacks was first proposed in the field of computer vision, including detection methods for model stealing attacks, surrogate model attacks, and evasion attacks [23–27]. Chen et al. [23] proposed a new adversarial sample defense method – stateful detection defense for image black-box attacks. Moreover, they proposed a similarity encoder based on the Euclidean distance metric. Then, they introduce a novel type of attack, query blinding, which is designed to bypass the stateful detection defense. This paper is evaluated using the CIFAR-10 dataset, and the experiments work well. However, this study applies well to image adversarial samples, but is limited to video classification, and does not involve malware detection.

Li et al. [24] designed Blacklight, a defense framework against query-based black-box adversarial attacks. The method uses probabilistic content fingerprint-based query matching to mitigate individual attack queries. They experimentally evaluated Blacklight on multiple datasets and image classification models for eight SOTA black-box attacks, and the experimental results were not only high in detection rate but also fast. Nevertheless, this method cannot defend against surrogate model attacks. If there are not highly similar adversarial examples, Blacklight can be evaded.

For deep neural network models (DNNs), Cohen et al. [26] put forward using Nearest-Neighbours and Influence Functions to detect adversarial samples. The core idea of this algorithm is that there should be a correspondence between the training data and the network classification. That is, for normal images, there is a strong correlation between their nearest neighbors in the DNN embedding space and their most helpful training examples, while adversarial examples are the opposite. They tested the performance of detection in both black-box and white-box attacks. However, this study uses the L_2 distance metric, which is computationally time-consuming and needs to be further improved in the future.

In order to quickly infer the intent of black-box attackers, Pang et al. [27] proposed a new estimation model, AdvMind. This model can reliably identify the query of interest (QOI) and accurately detect the target category of the attack at an early stage for timely remediation. The authors used four datasets and DNNs to perform experiments on the detection of three black-box attacks. However, AdvMind focuses on query-based attacks and is not effective for substitute model attacks.

With the increasing threat of black-box adversarial attacks in the industrial Internet of Things (IIOT), Esmaeili et al. [10] proposed a stateful query analysis strategy for the detection of adversarial scenarios. Their method includes two CNN-based components, namely similarity encoder, and classifier. Moreover, they introduced the Mahalanobis distance metric for the loss function of the detection model, which improved the detection rate. However, their architecture is to process the malware opcode features into greyscale images and use methods in the field of computer vision to classify and generate adversarial images, without generating malicious files. Future research on other distance indicators and data types should also be further developed.

As previous defense methods are static and cannot dynamically adapt to adversarial attacks, Li et al. [11] proposed the first instance-based online machine learning dynamic defense method against black-box attacks. Extensive experiments are conducted on image and malware datasets, and effects significantly outperform existing SOTA defense methods. Nevertheless, DyAdvDefender may need a manual inspection of samples to achieve optimal performance in the real world, and incorrect selection of malware feature sets may lead to defense failure.

Regarding a Windows adversarial attack, Fang et al. [8] proposed an automatic adversarial sample generation model based on reinforcement learning called RLAttackNet, which can successfully bypass the DeepDetectNet malware detection model. They proposed a new method for extracting features of PE files, including the Import Function Feature, General Information Feature, and Bytes Entropy Feature. Retraining the detection model by drawing on the idea of GAN revealed a significant decrease in the success rate of the attack. More attention needs to be paid to hyper-parameter optimization methods in deep learning models in the future.

In addition, unlike previous work, Maiorca et al. [9] presented a survey of PDF malware detection in an adversarial environment. They provide a comprehensive study on PDF preprocessing. Furthermore, they outline adversarial attacks against PDF malware detectors. They have discussed existing mitigating strategies as well as future research directions.

In summary, we can find that there are shortcomings in the existing research results in detecting malware black-box attacks: (1) Most researchers are devoted to the detection of image and PDF adversarial attacks, and the research on stateful detection of malware adversarial attacks is insufficient; (2) The existing stateful detection methods of malware black-box attacks need to extract features from original samples or convert them into images for further processing. Based on this, we propose the MalDBA method for stateful detection research on complete sample files.

3. Overview

3.1. Motivation

Machine learning has great potential in malware analysis, and DL-based malware detectors have been extensively studied, yet the problem remains unsolved. One of the key challenges currently facing malware detection and classification research is the adversarial examples [28]. Without addressing adversarial attacks, proposing malware detectors or classifiers is an endless and unfruitful task lacking substantial scientific advancement. For instance, the DL-based static malware detector proposed in another paper worked well in the evaluation, but malware adversarial samples still sneak through the model [13–15,29]. That is probably why the malware never stops despite the hundreds of detectors being proposed. It is urgent to detect black-box attacks based on the DL malware detector. The stateful detection method has been used in computer vision [11,23,24,27], but it has not been attempted in malware black-box attacks which generate real adversarial samples. Therefore, we designed the MalDBA framework to detect query-based black-box attacks. This article aims to detect the generation of adversarial samples, not to try to detect whether the input files are malicious or benign. When generating an adversarial sample, existing query-based black-box attacks produce a series of highly similar queries (i.e., each query in the set is similar to the previous queries), and the scores returned by the detector gradually change from malicious to benign. Based on this, we propose a defense approach that uses a similarity comparison algorithm to identify such queries and detects black-box attacks against malware detectors through this strategy.

3.2. Overall Framework

MalDBA mainly consists of four steps, namely training the malware detection model, simulating the black-box attack, saving the intermediate samples and prediction scores, and performing adversarial attack detection, as shown in Figure 2.

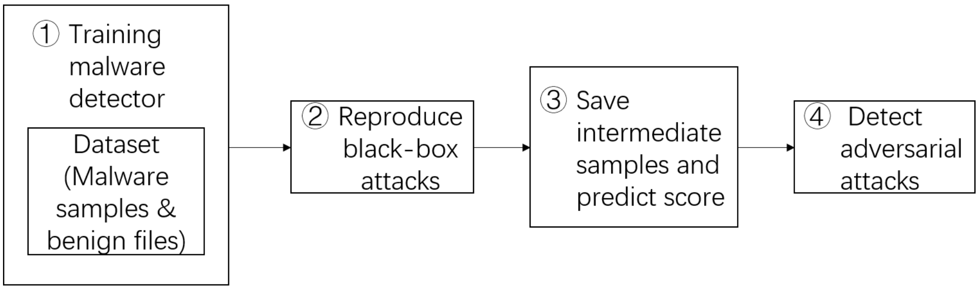

① Training malware detector
Dataset (Malware samples & benign files)
② Reproduce black-box attacks
③ Save intermediate samples and predict score
④ Detect adversarial attacks

Figure 2. The process of MalDBA.

4. Our Scheme

4.1. Training Malware Detector

The function of this step is to train a mature malware detection model. For the DL-based static malware detector, we choose the MalConv model(as shown in Figure 3), which is not only the current popular malware detection model, but also the target model selected by many malware adversarial attacks [14,15,30–35]. By training the MalConv model, a binary classifier that can distinguish benign samples from malicious samples can be obtained.

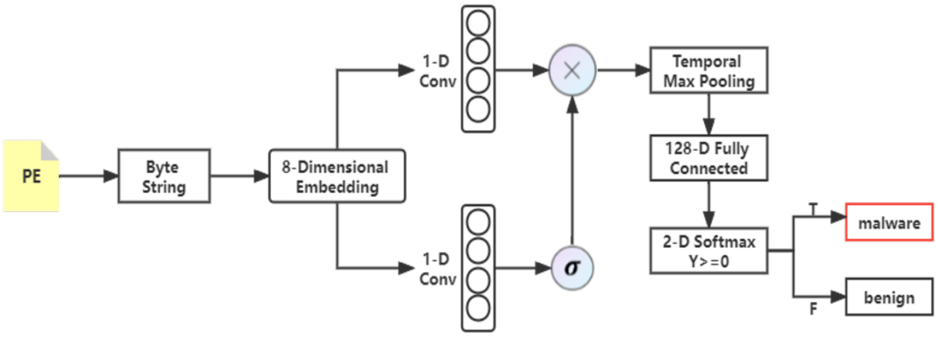


Figure 3. The architecture of MalConv model.

MalConv model is the first convolutional neural network architecture (CNN) addressing the classification problem of extremely long sequences, proposed by Raff et al. [13]. Its input is a PE file and returns a score to judge whether this file is malware or not. The model distinguishes programs based on the byte representation of the input, without extracting any features. If the input file length exceeds 2MB, the file will be truncated to the specified size; otherwise, the file will be padded with the value 0.

4.2. Reproduce Black-Box Attacks

Our work is dedicated to detecting query-based black-box attacks and the function of this module is to reproduce typical query-based black-box attacks. Since malware adversarial attacks were investigated later than image adversarial attacks and most of the AEs are generated on feature vectors or substitute models [28,36–40], there are not many query-based black-box attack methods that can generate real AE files and publish open source codes [14,15,41,42]. We choose two advanced score-based black-box attack frameworks [14,15]. The target detectors of these two attacks are both MalConv models, we reproduce them through open-source code and compare the attack success rate. The process of generating adversarial samples is roughly illustrated in Figure 4.

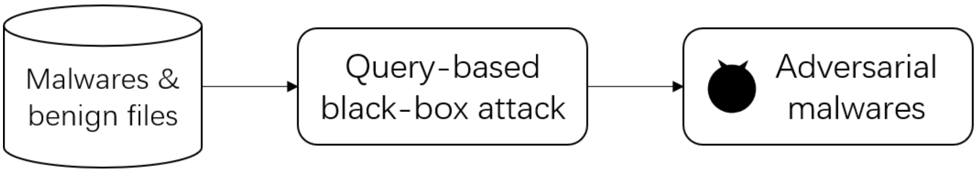


Figure 4. The process of generating adversarial examples.

4.3. Save the Intermediate Samples

Figure 5 shows the process of saving the historical query results of the black-box attack. The historical queries (i.e., intermediate samples), as well as the prediction scores returned from the detector in the process of generating adversarial sample queries, are saved in preparation for the next step.

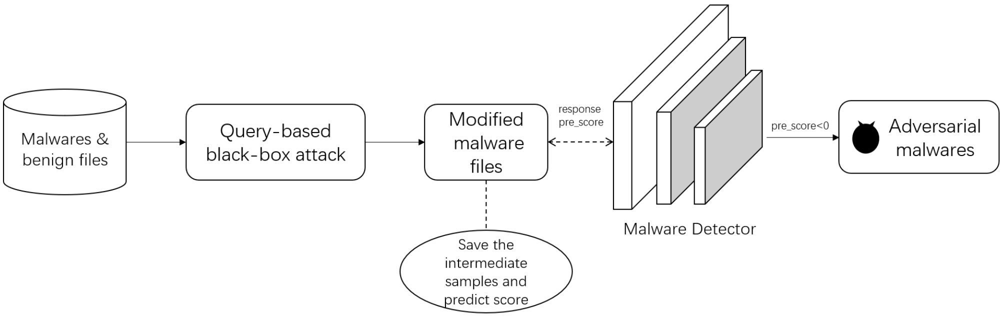


Figure 5. The process of saving the history queries.

4.4. Detect the Adversarial Attacks

Algorithm 1 sketches the procedure of MalDBA. Different numbers of benign and malicious samples are randomly selected with the intermediate samples saved above to form the indexed sample sets of different sizes. Then use the similarity comparator based on the Minhash algorithm to compare the similarity and judge whether the scores returned by MalConv gradually decrease, so as to determine whether there is an adversarial attack and achieve the purpose of defense. The process of detection is shown in Figure 6.

```
Algorithm 1: The procedure of MalDBA
Initialization: indexed samples set $K$, query_set $(q_1, \ldots, q_n)$, $predict\_score\ S$,
        similarity comparator H. $(q_i \in K)$
Output: Whether adversarial attacks exist in the $K$ (Ture or False)
for $c$ in $K$ do
    $L_c = new\ List\ ()$
    $L_c \leftarrow Obtain\ the\ index\ set\ of\ samples\ similar\ to\ c\ through\ H$
end for
if $(q_1, \ldots, q_n)$ $in\ the\ same\ L\ and$ $(S_{q_1}, \ldots, S_{q_n})$ $decline$ then
return True
```

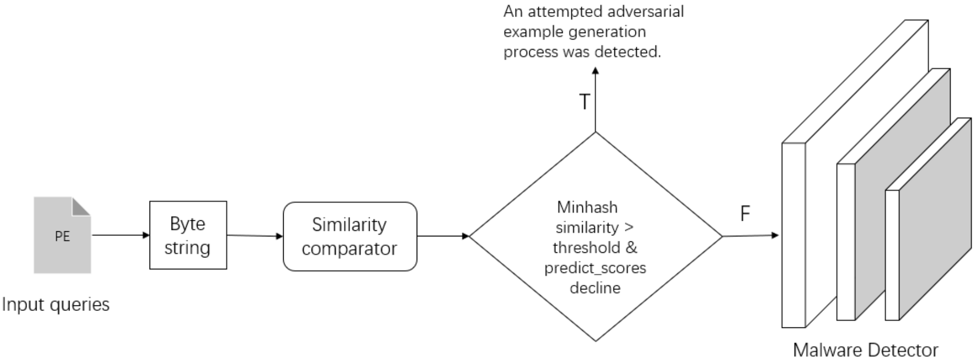

An attempted adversarial example generation process was detected.
T
PE
Input queries
Byte string
Similarity comparator
Minhash similarity > threshold & predict_scores decline
F
Malware Detector

Figure 6. The process of detection.

5. Evaluation

5.1. Experimental Setup

We implemented MalDBA in Python. The experimental environment is configured as follows: (1) Lenovo ThinkStation, Intel®Core TM i7-6700U CPU @3.40GHz × 16.0 GB RAM, and an Nvidia GeForce GTX 1070 (2) 64bit Windows 10 operation system, (3) Pycharm Professional Edition with Anaconda plugin 2020.

5.1.1. Dataset

The experimental data in this paper includes malware samples and benign files, among which malicious samples are from VirusShare corpus [12], and benign PE files are extracted from Windows 10 system files and different software companies. Since the input file size of the GAMMA model cannot exceed 1MB, we filtered the dataset (samples larger than 1MB are only a minority). Table 1 and Figure 7 illustrates the distribution of the dataset.

Table 1. The Dataset.

Dataset	Benign Files	Malicious Files	Total
Num	5309	3720	9029

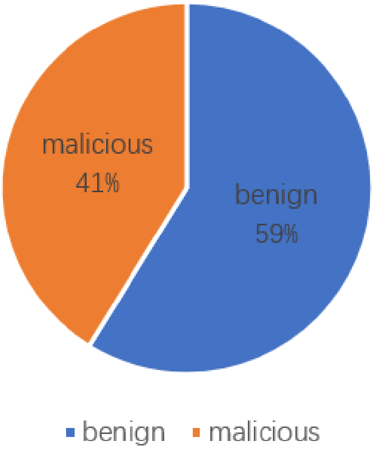


Figure 7. The distribution of a dataset.

5.1.2. Black-Box Attack Methods

Two typical query-based malware black-box attacks are selected for experimentation during our evaluation (as shown in Table 2).

Table 2. Query-based malware black-box attacks.

Black-Box Attack	Method
MalRNN [14]	MalRNN automatically generates adversarial examples to attack DL-based static malware detectors in the way of language modeling. Using the Seq2Seq RNN Language Model to generate benign looking byte sequences successfully eludes anti-malware engines.
GAMMA [15]	GAMMA is a malware adversarial attack method based on optimized genetic algorithm. It extracts benign contents which are easy to evade the DL-based static malware detector and injects them into the end of malicious samples or the newly-created sections (i.e., Padding and Section-Injection attacks)

5.2. Experimental Results and Discussion

In this section, we experimentally evaluate the detection effectiveness of MalDBA. Firstly, we evaluate the MalConv malware detector model selected using the original dataset. Secondly, the selected black-box attack algorithm is applied to the dataset and target detector, and then our proposed MalDBA method is used to detect the black-box attacks and evaluate the attack success rate of the attacks without and with the defense. After that, the relationship between the average response time (ART) and attack detection rate (ADR) with the number of indexed samples (K) on attacks is discussed. Finally, we compare the MalDBA with similar studies.

5.2.1. The Experimental Results of Malware Detector

We chose MalConv, a popular DL-based static malware detection model, which is used as the target model for many malware adversarial attacks [14,15,30–35]. We reproduced the model using the Python programming language. The dataset is divided according to the ratio of training set: validation set: test set = 6:2:2. We conduct the experiments on randomly partitioned datasets and the results are shown in Table 3. The accuracy of the test set is 95.03%, which is not far from the experimental results of the original paper [13].

Table 3. Performance of MalConv model.

Metrics Detector	Test_Loss	Test_Accuracy	Train_Loss	Train_Accuracy
MalConv	0.1380	95.03%	0.0862	96.32%

5.2.2. The Detection Results with Different Black-Box Attack Methods

In this section, we replicate the MalRNN and GAMMA black-box attack frameworks, using Attack Success Rate (ASR) as an evaluation metric. Each experiment is performed three times, and the results are averaged as the final experimental results. As shown in Table 4, the effectiveness of two black-box attacks with no defense and defense with the MalDBA detection method is presented. It can be seen from Table 4 that our defense method can reduce the success rate of the attacks to 0%.

Table 4. Attack success rate (ASR) of attacks.

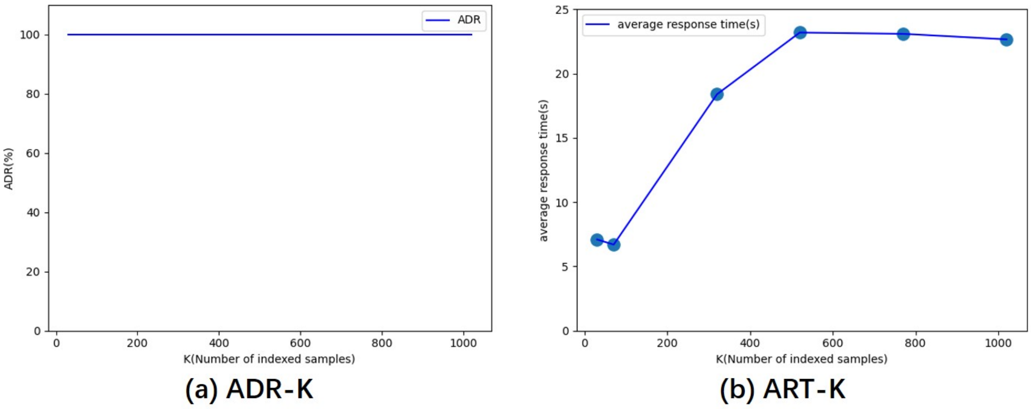
Attack	Defence	ASR
MalRNN	No defence	88.6%
	MalDBA	0%
GAMMA	No defence	86.3%
	MalDBA	0%

5.2.3. The Relation between the ART and ADR with K on Attacks

We randomly save 20 historical query results of malware and randomly select different numbers of benign and malicious files respectively to form the indexed sample set K. The sizes of *K* are taken as 30, 70, 320, 520, 770, and 1020, respectively. The relationship between the average response time (ART) and the number of indexed samples (*K*) for MalRNN and GAMMA attacks are shown in Table 5. The relationship between the ART, attack detection rate (ADR) with *K* for these two attacks are depicted in Figures 8 and 9 respectively. From the figures, it can be found that the ADR of MalDBA for these two black-box attacks is 100%, and the ADR is independent of *K*. With increasing *K*, the ART fluctuates to a certain extent and then gradually stabilizes around 23 s.

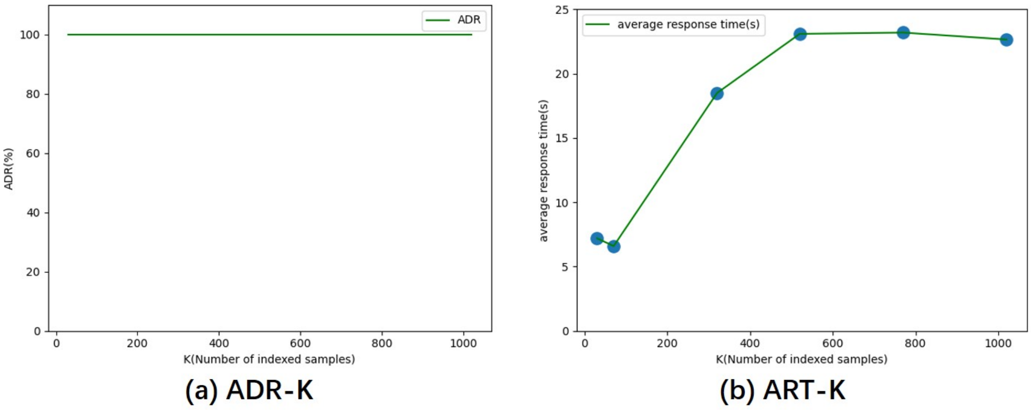
Table 5. The relationship between the average response time (ART) and the number of indexed samples (*K*) on MalRNN and GAMMA.

ART(s) *K* **Attack**	**30**	**70**	**320**	**520**	**770**	**1020**
MalRNN	7.11	6.71	18.41	23.20	23.10	22.67
GAMMA	7.20	6.62	18.50	23.11	23.22	22.72



(a) ADR-K (b) ART-K

Figure 8. The relation between the attack detection rate (ADR) and average response time (AST) with the number of indexed samples (*K*) on MalRNN attack.



(a) ADR-K (b) ART-K

Figure 9. The relation between the attack detection rate (ADR) and average response time (AST) with the number of indexed samples (*K*) on GAMMA attack.

5.2.4. Comparison with Similar Studies

In this section, we compare MalDBA with similar studies in terms of datasets, target models, experimental setup, the accuracy of the target model, and attack detection rate (ADR). The results of the comparison are shown in Table 6.

Table 6. Comparison with similar studies.

	MalDBA	**Esmaeili et al. [10]**	**Miles Q. Li et al. [11]**	**Steven Chen et al. [23]**	**Ren Pang et al. [27]**	**Huiying Li et al. [24]**
Datasets	PE	2-digit hexadecimal bytecode vectors	CIFAR-10, MNIST, PE	CIFAR-10	CIFAR-10, CIFAR-100, ISIC, Mini-VGGface2	MNIST, GTSRB, CIFAR10, ImageNet
Target models	Malconv	CNN	CNN, FNN	ResNet	DNNs	DNN
Experimental setup	A desktop with one Intel$^{\circledR}$Core $^{\text{TM}}$ i7-6700U CPU, 16.0 GB RAM , and an Nvidia GeForce GTX 1070	-	A server with one Intel$^{\circledR}$Core $^{\text{TM}}$ i9-9980XE CPU, 128 GB memory, and an Nvidia GeForce RTX 2080 Ti Graphics Card	-	-	Nvidia Titan RTX
Accuracy of target model	95.03%	98%	-	92%	92.44%, 70.47%, 88.17%, 96.17%	-
ADR	100%	93.1%	Extract PE Strings feature: 94.7%	100%	Can reach to 100%	Can reach to 100%

Compared with similar studies, MalDBA has the following advantages: (1) MalDBA references the idea of image stateful detection, but does not need to convert PE files into images (which will lose some important features). (2) MalDBA can directly detect complete malware, skipping the dataset preprocessing, feature extraction, feature selection, and feature fusion stages, saving a lot of time. (3) MalDBA requires a moderate-performance hardware platform, so it has good universality and a high detection rate.

6. Discussion

Our proposed detection method operates on complete files, which inevitably takes some time. Therefore, we put forward an idea: drawing on the knowledge of computer vision, extracting the features of the deep neural network model's middle layer for sample similarity comparisons in order to detect adversarial attacks [43,44]. We adopted three methods to carry out experiments with different numbers of indexed samples (K). The MalConv was chosen for the deep neural network model and the MalRNN framework was selected as the black-box attack model. After extracting the features of the neural network model's middle layer, we adopted L_2 distance, K-means, and Minhash methods to measure the similarity among the indexed samples. Experimental results of different methods with different numbers of index samples are shown in Table 7. From the table, it can be seen that the features of the neural network model's middle layer are not effective for the similarity measure among the samples. The existence of an adversarial attack could not be detected. The reason for this may be that PE samples and images are fundamentally different: The middle layer features of an image under a deep neural network model is an image whose general outline can still be seen, whereas the middle layer features of a malicious or benign sample is a multidimensional array of tensors.

Table 7. The effects of different methods with the number of indexed samples (K) under MalRNN.

K **Methods**	**30**	**200**	**400**
L_2	×	×	×
K-means	×	×	×
Minhash	×	×	×

'×' denotes the features of the neural network model's middle layer are ineffective for the similarity measure among the samples.

7. Limitations and Conclusions

Limitations of MalDBA: (1) The false positive rate of the MalDBA will rise if highly similar malicious samples are fed into the detector for querying (as if there is a similarity among malicious samples of the same family). (2) MalDBA detects historical query sequences generated during the iteration of query-based black-box attacks and cannot defend against non-query-based attacks (e.g., substitute model attacks).

Malware black-box attacks cause security risks to AI and pose a threat to data security as well as privacy, and their defense is a complex issue [18,19]. In this paper, we manage to solve the problem of stateful detection for malware score-based black-box attacks. First, the set of historical query samples generated during the attack is saved. Afterward, similarity comparison is performed on different numbers of indexed samples by a similarity comparator. Finally, the presence or absence of an adversarial attack is detected according to the trend of scores returned by the malware detector. The results show that the detection rate of MalDBA against score-based black-box attacks is 100%, and the detection rate is independent of the number of indexed samples.

In the future, we plan to investigate the following research directions: (1) Study of a general attack strategy for stateful detection defense. (2) Drawing on the similarity encoder proposed in computer vision, consider whether it can be studied by extracting the function call graph or control flow graph of malware and combining it with graph neural networks.

Author Contributions: Conceptualization, Z.L.; Funding acquisition, J.X.; Methodology, Z.K.; Validation, Y.W.; Writing—original draft, W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by China Scholarship Council (No. 202106030119), the National Natural Science Foundation of China (No. 62172042), and Major Scientific and Technological Innovation Projects of Shandong Province (2020CXGC010116).

Data Availability Statement: Previously published articles were used to support this study and these prior studies and datasets are cited at relevant places within this article. The link to the datasets is https://virusshare.com/ (accessed on 6 February 2022).

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Mcafee. *Labs Threats Report*; McAfee: Hong Kong, China, 2021; 24p.
2. VirusTotal. Deception at Scale: How Malware Abuses Trust; VirusTotal: Dublin, Ireland, 2022; 15p.
3. Darem, A.; Abawajy, J.; Makkar, A.; Alhashmi, A.; Alanazi, S. Visualization and deep-learning-based malware variant detection using OpCode-level features. *Future Gener. Comput. Syst.* **2021**, *125*, 314–323. [CrossRef]
4. Sun, G.; Qian, Q. Deep learning and visualization for identifying malware families. *IEEE Trans. Dependable Secur. Comput.* **2018**, *18*, 283–295. [CrossRef]
5. Huang, X.; Ma, L.; Yang, W.; Zhong, Y. A method for windows malware detection based on deep learning. *J. Signal Process. Syst.* **2021**, *93*, 265–273. [CrossRef]
6. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
7. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
8. Fang, Y.; Zeng, Y.; Li, B.; Liu, L.; Zhang, L. DeepDetectNet vs RLAttackNet: An Adversarial Method to Improve Deep Learning-Based Static Malware Detection Model. *PLoS ONE* **2020**, *15*, e0231626. [CrossRef]
9. Maiorca, D.; Biggio, B.; Giacinto, G. Towards Adversarial Malware Detection: Lessons Learned from PDF-based Attacks. *ACM Comput. Surv.* **2020**, *52*, 1–36.
10. Esmaeili, B.; Azmoodeh, A.; Dehghantanha, A.; Zolfaghari, B.; Karimipour, H.; Hammoudeh, M. IIoT Deep Malware Threat Hunting: From Adversarial Example Detection to Adversarial Scenario Detection. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8477–8486. [CrossRef]
11. Li, M.Q.; Fung, B.C.; Charland, P. DyAdvDefender: An instance-based online machine learning model for perturbation-trial-based black-box adversarial defense. *Inf. Sci.* **2022**, *601*, 357–373. [CrossRef]
12. Available online: https://virusshare.com/ (accessed on 6 February 2022).
13. Raff, E.; Barker, J.; Sylvester, J.; Brandon, R.; Catanzaro, B.; Nicholas, C.K. Malware detection by eating a whole exe. In Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
14. Ebrahimi, M.; Zhang, N.; Hu, J.; Raza, M.T.; Chen, H. Binary Black-box Evasion Attacks Against Deep Learning-based Static Malware Detectors with Adversarial Byte-Level Language Model. In Proceedings of the 2021, AAAI Workshop on Robust, Secure and Efficient Machine Learning (RSEML), Vancouver, BC, Canada, 2–9 February 2021.

15. Demetrio, L.; Biggio, B.; Lagorio, G.; Roli, F.; Armando, A. Functionality-preserving black-box optimization of adversarial windows malware. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3469–3478. [CrossRef]
16. Wu, W.; Li, B.; Chen, L.; Gao, J.; Zhang, C. A review for weighted minhash algorithms. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 2553–2573. [CrossRef]
17. Podschwadt, R.; Takabi, H. On effectiveness of adversarial examples and defenses for malware classification. In *International Conference on Security and Privacy in Communication Systems*; Springer: Cham, Switzerland, 2019; pp. 380–393.
18. Li, D.; Li, Q. Adversarial deep ensemble: Evasion attacks and defenses for malware detection. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3886–3900. [CrossRef]
19. Huang, Y.; Verma, U.; Fralick, C.; Infantec-Lopez, G.; Kumar, B.; Woodward, C. Malware evasion attack and defense. In Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Portland, OR, USA, 24–27 June 2019; pp. 34–38.
20. Li, H.; Xu, X.; Zhang, X.; Yang, S.; Li, B. Qeba: Query-efficient boundary-based blackbox attack. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1221–1230.
21. Brendel, W.; Rauber, J.; Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv* **2017**, arXiv:1712.04248.
22. Yoon, J.; Hwang, S.J.; Lee, J. Adversarial purification with score-based generative models. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 12062–12072.
23. Chen, S.; Carlini, N.; Wagner, D. Stateful detection of black-box adversarial attacks. In Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence, Virtual Event, 13 November 2020; pp. 30–39.
24. Li, H.; Shan, S.; Wenger, E.; Zhang, J.; Zheng, H.; Zhao, B.Y. Blacklight: Scalable defense for neural networks against query-based black-box attacks. *arXiv* **2022**, arXiv:2006.14042.
25. Juuti, M.; Szyller, S.; Marchal, S.; Asokan, N. PRADA: Protecting against DNN model stealing attacks. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 17–19 June 2019; pp. 512–527.
26. Cohen, G.; Sapiro, G.; Giryes, R. Detecting adversarial samples using influence functions and nearest neighbors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14453–14462.
27. Pang, R.; Zhang, X.; Ji, S.; Luo, X.; Wang, T. AdvMind: Inferring adversary intent of black-box attacks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 1899–1907.
28. Al-Dujaili, A.; Huang, A.; Hemberg, E.; O'Reilly, U.M. Adversarial deep learning for robust detection of binary encoded malware. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 76–82.
29. Castro, R.L.; Schmitt, C.; Dreo, G. Aimed: Evolving malware with genetic programming to evade detection. In Proceedings of the 2019 18th IEEE International Conference On Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference On Big Data Science and Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 5–8 August 2019; pp. 240–247.
30. Luca, D.; Biggio, B.; Giovanni, L.; Roli, F.; Alessandro, A. Explaining vulnerabilities of deep learning to adversarial malware binaries. In Proceedings of the 3rd Italian Conference on Cyber Security, ITASEC 2019, Pisa, Italy, 12 February 2019; Volume 2315.
31. Kolosnjaji, B.; Demontis, A.; Biggio, B.; Maiorca, D.; Giacinto, G.; Eckert, C.; Roli, F. Adversarial malware binaries: Evading deep learning for malware detection in executables. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 533–537.
32. Mosli, R.; Slota, T.J.; Pan, Y. Creating Adversarial Malware Examples Through Guided Metamorphic Changes. In Proceedings of the 2021 IEEE International Symposium on Technologies for Homeland Security (HST), Boston, MA, USA, 8–9 November 2021; pp. 1–7. [CrossRef]
33. Quertier, T.; Marais, B.; Morucci, S.; Fournel, B. MERLIN—Malware Evasion with Reinforcement LearnINg. *arXiv* **2022**, arXiv:2203.129802022.
34. Dasgupta, P.; Osman, Z. A Comparison of State-of-the-Art Techniques for Generating Adversarial Malware Binaries. *arXiv* **2021**, arXiv:2111.11487.
35. Burr, J.; Xu, S. Improving Adversarial Attacks Against Executable Raw Byte Classifiers. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10–13 May 2021; pp. 1–2. [CrossRef]
36. Li, X.; Nie, Y.; Wang, Z.; Kuang, X.; Qiu, K.; Qian, C.; Zhao, G. BMOP: Bidirectional Universal Adversarial Learning for Binary OpCode Features. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8876632. [CrossRef]
37. Rosenberg, I.; Shabtai, A.; Rokach, L.; Elovici, Y. Generic black-box end-to-end attack against state of the art API call based malware classifiers. In *International Symposium on Research in Attacks, Intrusions, and Defenses*; Springer: Cham, Switzerland, 2018; pp. 490–510.
38. Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; McDaniel, P. Adversarial perturbations against deep neural networks for malware classification. *arXiv* **2016**, arXiv:1606.04435.
39. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef]
40. Hu, W.; Tan, Y. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv* **2017**, arXiv:1702.05983.

41. Yuste, J.; Pardo, E.G.; Tapiador, J. Optimization of code caves in malware binaries to evade machine learning detectors. *Comput. Secur.* **2022**, *116*, 102643. [CrossRef]
42. Demetrio, L.; Coull, S.E.; Biggio, B.; Lagorio, G.; Armando, A.; Roli, F. Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. *ACM Trans. Priv. Secur. (TOPS)* **2021**, *24*, 1–31. [CrossRef]
43. Sünderhauf, N.; Dayoub, F.; Shirazi, S.; Upcroft, B.; Milford, M. On the Performance of ConvNet Features for Place Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015.
44. Qiao, Y.; Cappelle, C.; Ruichek, Y.; Yang, T. ConvNet and LSH-based visual localization using localized sequence matching. *Sensors* **2019**, *19*, 2439. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.