Model Evasion Attacks Against Partially Encrypted Deep Neural Networks in Isolated Execution Environment

Kota Yoshida(✉) and Takeshi Fujino

Ritsumeikan University, Shiga, Japan
ri0044e@ed.ritsumei.ac.jp, fujino@se.ritsumei.ac.jp

Abstract. It is important to hide DNN models from adversaries not only for protecting intellectual property but also for preventing attacks. Isolated execution environments (IEEs) are necessary to protect the DNN model in an edge device. Conventional studies on DNN inference in IEEs have focused on model confidentiality and not the threat of model evasion attacks. If some of the model parameters or intermediate results are leaked to adversaries, model evasion attacks may occur even if the confidentiality of the model is secured. In this work, we performed attacks against partially encrypted DNN models that are executed on IEEs. In an existing proposal, a feature extractor of the model is executed in a normal world and a classifier is executed in a secure enclave, but there is still a threat that an adversary may perform a gradient-based model evasion attack against a feature extractor. We performed gradient-based model evasion attacks against the feature extractor more efficiently by preparing multiple guide images. Our results clarified that all parameters on the feature map should be kept secret by the parameter encryption. In addition, we consider another risk case where calculated values on the feature extractor are stored in the unencrypted memory and demonstrated the gradient estimation-based model evasion attack by exploiting the intermediate feature maps. Our results indicate that both DNN model parameters and intermediate feature maps should be concealed not only for protecting intellectual property but also for preventing model evasion attacks.

Keywords: Model evasion attack · Adversarial examples · Trusted execution environment · Deep neural network

1 Introduction

Trained deep neural networks (DNNs) are widely used in various tasks such as image recognition. DNNs require high costs for training, including the domain knowledge to optimize the model architecture, a huge dataset that is annotated by hand, and computing the resources to calculate parameter optimization. Therefore, a trained DNN model that is implemented in commercial services is quite valuable.

J. Zhou et al. (Eds.): ACNS 2021 Workshops, LNCS 12809, pp. 78–95, 2021.
https://doi.org/10.1007/978-3-030-81645-2_6

Information of DNN models, especially model architectures and parameters, is also valuable to adversaries. In a white-box scenario, which assumes an adversary knows information about the DNN models, various threats have been reported. The model inversion attack estimates training data that include confidentiality and privacy from DNN models. The model evasion attack calculates a perturbation and then adds it to an input, which is called an adversarial example. This adversarial example causes misrecognition.

It is important to hide DNN models from adversaries not only for protecting intellectual property but also for preventing attacks. Homomorphic encryption [5,18], multi-party computing [9], and isolated execution [6,13] all allow calculations while hiding the DNN model from adversaries. While isolated execution is a promising choice in an edge device, isolated execution environments (IEEs) have limited computing resources and it is difficult to unroll all operations of the DNN model inferences onto an enclave. Hanzlik et al. [6] proposed dividing the DNN model inference operations into individual layers and then executing them layer-by-layer on the enclave, but the calculation time is longer than in a normal environment. Schlogl et al. proposed eNNclave, where transfer learning enables splitting the DNN model into normal (feature extractor) and confidential (classifier) parts. The normal part of the DNN model is executed in the normal world, and the confidential part is executed in the secure enclave. This enables faster inference because it can use the abundant computational resources in the normal world (e.g., GPUs). While promising, these studies have focused on model confidentiality and did not consider model evasion attacks. Model evasion attacks are possible without all of the information about models. As such, attacks may occur when a part of the information is leaked even if it is not enough to restore the entire model.

In this paper, we focus on IEEs and model evasion attacks. We investigate which information of the DNN model inference should be hidden from adversaries to prevent the attacks by using IEEs. We assume that a feature extractor is leaked if it is executed in the normal world. We evaluate gradient-based model evasion attacks against the feature extractor on an eNNcalve scenario. We assume that intermediate feature maps are leaked if they are stored in the normal world memory in plain text when an enclave switches the processing layer. We evaluate gradient estimation-based model evasion attacks with the intermediate feature maps.

Our contributions are as follows.

- We studied model evasion attacks against DNN models that are partially executed on IEEs.
- We found a more efficient method for gradient-based model evasion attacks against a feature extractor by preparing multiple guide images. The guide images mean that images acquired from the guide (the adversary's misclassification target) class.
- We performed gradient estimation-based model evasion attacks with an intermediate feature map. In this scenario, all model parameters are hidden but the calculated intermediate values (i.e., feature maps) are accessible by the

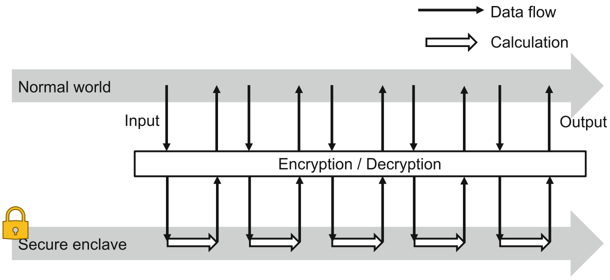


Fig. 1. Overview of MLCapsule [6].

adversary. The experimental results indicate that the intermediate values, as well as the model parameters, should be hidden from the adversary.

2 Related Works

2.1 DNN Model Inference on IEEs

Hanzlik et al. [6] proposed MLCapsule, where a DNN model provider protects the intellectual property of ML services and clients perform inference offline. Figure 1 shows an overview of MLCapsule. All DNN model parameters, such as weight and bias, are encrypted. The user interacts with the provider and sets up a secure enclave. A part of the layers is loaded and decoded on the enclave, and inference is performed. The internal state of the DNN inference is encrypted and stored in the normal world. The user repeats operations for each layer and obtains the result of the inference. Generally, the enclave can not use an inference accelerator, and cryptographic operations cause an increase in processing time.

Schlogl et al. proposed eNNclave [13] for improving DNN model inference speed on IEEs. eNNclave assumes that a DNN model is trained by transfer learning, an overview of which is shown in Fig. 2. Transfer learning diverts a feature extractor of a trained public model to other tasks. It is performed in the following steps.

1. Extract a feature extractor from a trained public model.
2. Add a classifier to the feature extractor and initialize.
3. Fix the parameters of the feature extractor and train the classifier.

Note that the feature extractor of the transferred model is the same as the public model's one. Figure 3 shows an overview of eNNclave. In this scenario, secret information on the target task after transfer learning is only included in the classifier. Schlogl et al. claimed that the feature extractor of the transferred model was public information and the transferred model information could be sufficiently hidden by executing only the classifier in the secure enclave. The feature etractor can be executed in the normal world without encryption and accelerated by GPUs.

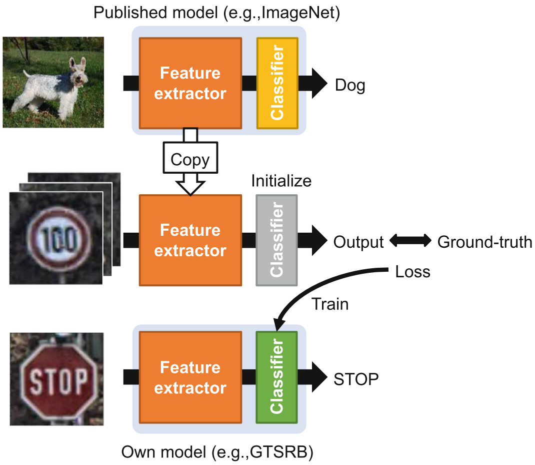


Fig. 2. Overview of transfer learning.

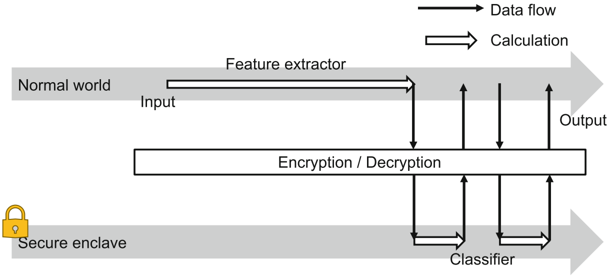


Fig. 3. Overview of eNNclave [13].

2.2 Model Evasion Attacks

The purpose of model evasion attacks on image classification tasks is to calculate perturbations that cause misclassification. An input image with perturbation is called an adversarial example. Scenarios in which an adversary specifies a misclassification target class are called a targeting attack.

If an adversary knows all target model architecture and parameters, it can calculate an adversarial example by calculating a gradient of an input (source) image with respect to an adversarial target (guide) class. This is a white-box scenario. In this paper, we call such an attack a gradient-based model evasion attack. The fast gradient sign method (FGSM) [4] and the momentum iterative FGSM (MI-FGSM) [3], which is an improved version of FGSM, are typical methods of the model evasion attack. An overview of the gradient-based attack is shown in Fig. 4. Adversarial examples by targeted MI-FGSM (TMI-FGSM) are calculated by the following equations:

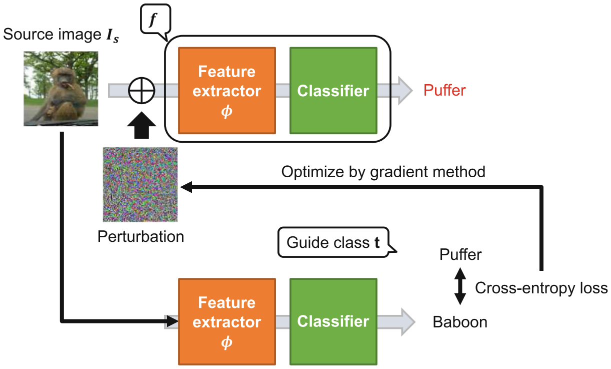


Fig. 4. Overview of gradient-based model evasion attack.

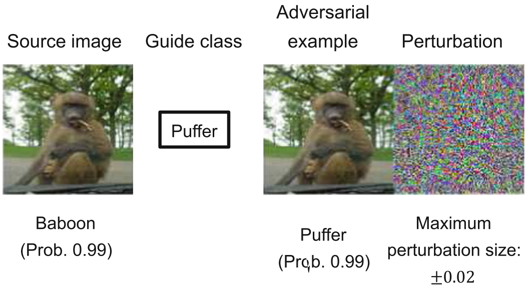


Fig. 5. Example of TMI-FGSM [3] attack on image classifier trained by ImageNet.

$$\begin{aligned} I_{adv}^0 &= I_s \\ I_{adv}^{k+1} &= clip(I_{adv}^k - \alpha \times sign(m^{k+1})), \end{aligned} \tag{1}$$

$$\begin{aligned} m^0 &= 0 \\ m^{k+1} &= m^k + \frac{\nabla_{I_{adv}^k} J(f(I_{adv}^k), t)}{||J(f(I_{adv}^k), t)||_1}, \end{aligned} \tag{2}$$

where I_{adv} is an adversarial example, I_s is a source image, t is an adversarial target class, k is the number of iterations, α is the step size, function $clip(a)$ clips input a into $a \in [0, 1]$, function $sign(a)$ calculates the sign of input a, function $f(a)$ is a DNN model that calculates the classification result from input image a, and function $J(a, b)$ calculates the cross-entropy between a and b. A result of a TMI-FGSM attack against an image classifier trained by ImageNet is shown in Fig. 5.

In the gradient-based model evasion attack, an adversary does not necessarily need to know all the architecture and parameters of the target model. If an adversary knows only the architecture and parameters of the feature extractor

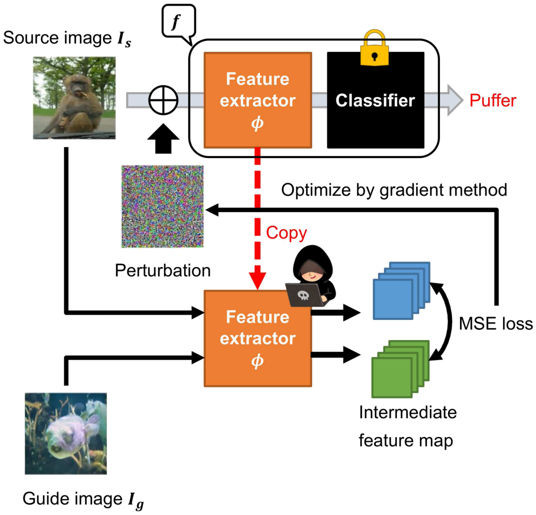


Fig. 6. Overview of gradient-based model evasion attack against feature extractor.

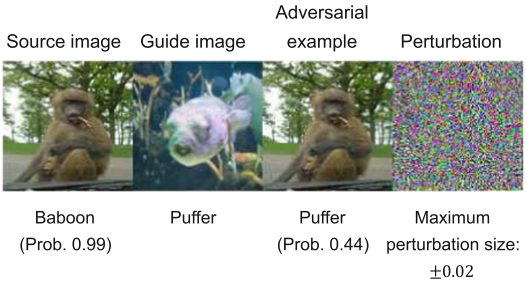


Fig. 7. Example of TMI-FGSM against feature extractor from an image classifier trained by ImageNet.

of the target model, it can calculate an adversarial example by calculating a gradient of the source image with respect to the distance between the intermediate feature map of the source and guide images [7,8,12,17]. The guide images mean that images acquired from the guide (the adversary's misclassification target) class. In this attack, instead of improving the probability of the guide class, the adversary calculates the perturbation to bring the intermediate representation (feature map) of the source image closer to the intermediate representation of the guide image. In this paper, we call this attack a gradient-based model evasion attack against a feature extractor. Wang et al. pointed out that a DNN model trained with transfer learning is vulnerable to this attack [17]. Inkawhich et al. extended TMI-FGSM to against feature extractor [8]. An adversarial example by the extended TMI-FGSM is calculated by replacing Eq. 2 with Eq. 3 (Fig. 6):

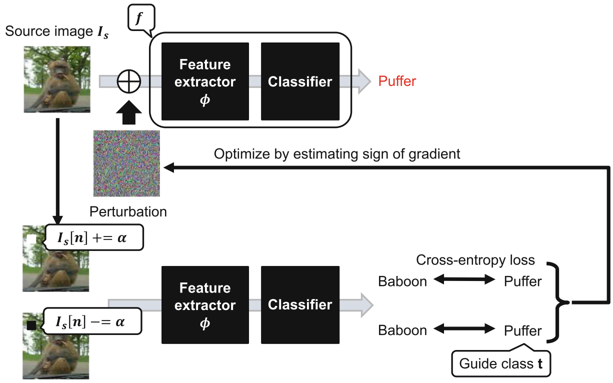


Fig. 8. Overview of black-box attack with class probability.

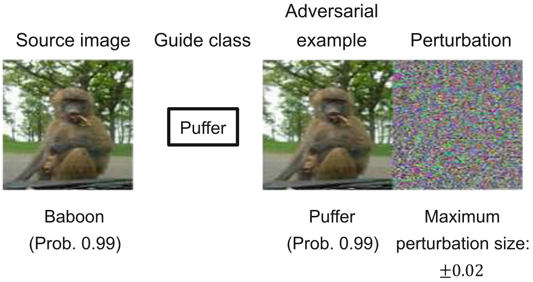


Fig. 9. Example of SimBA [10] on image classifier trained by ImageNet.

$$
\begin{aligned}
m^0 &= 0 \\
m^{k+1} &= m^k + \frac{\nabla_{I_{adv}^k} MSE(\phi(I_{adv}^k), \phi(I_g))}{||MSE(\phi(I_{adv}^k), \phi(I_g))||_1},
\end{aligned}
\tag{3}
$$

where function $MSE(a, b)$ calculates a mean squared error between a and b, function $\phi(a)$ is a feature extractor that calculates an intermediate feature map by input image a, and I_g is a guide image that belongs to a guide class. A result of a TMI-FGSM attack against feature extractor of an image classifier trained by ImageNet is shown in Fig. 7.

If an adversary does not know the target model architecture or parameters but has access to the classification result, it can calculate an adversarial example by estimating a gradient of a source image with respect to a guide class probability [1,10,14]. In this paper, we call this attack a gradient estimation-based model evasion attack. The adversary estimates the gradient of the source image from the change in output probability when the source image is changed slightly. A simple black-box attack (SimBA) [10] is a typical method of the model evasion attack. Algorithm 1 is an algorithm of SimBA. Figure 8 shows an overview of

Algorithm 1. Simple black-box attack (SimBA) [10]

```
Input: Source image $I_s$, Guide class $t$, Number of image pixels $N$,
  Perturbation parameter $\alpha$, DNN model $f$, Cross-entropy function $J$
Output: Adversarial example $I_{adv}$
  $I_{adv} = I_s$
  for n=0 to N-1 do
    $x' = I_{adv}$
    $x'[n] = I_{adv}[n] + \alpha$
    $p^+ = J(f(x'), t)$
    $x' = I_{adv}$
    $x'[n] = I_{adv}[n] - \alpha$
    $p^- = J(f(x'), t)$
    if $p^+ < p^-$ then
      $I_{adv}[n] = I_{adv}[n] + \alpha$
    else if $p^+ > p^-$ then
      $I_{adv}[n] = I_{adv}[n] - \alpha$
    else
      $I_{adv}[n] = I_{adv}[n] + 0$
    end if
    $I_{adv} = clip(I_{adv})$
  end for
  return $I_{adv}$
```

Table 1. Related model evasion attacks and this work.

Gradient-based		Gradient estimation-based	
Whole model	Feature extractor	Class probability	Intermediate feature-map
FGSM [4] MI-FGSM [3]	Sabour et al. [12] Wang et al. [17] Huang et al. [7] Extended MI-FGSM [8] Extended MI-FGSM with multiple guide images (Sect. 3)	SimBA [10] Bhagoji et al. [1] Senzaki et al. [14]	Extendetd SimBA (Sect. 5)

the targeted SimBA. A result of the SimBA against an image classifier trained by ImageNet is shown in Fig. 9.

In this paper, we improve the TMI-FGSM against a feature extractor [8] by using multiple guide images in Sect. 3. In Sect. 4, we evaluate the TMI-FGSM against a feature extractor on eNNclave scenario [13]. In Sect. 5, we extend SimBA from class probability-based to intermediate feature map-based and evaluate the attack in the MLCapsule scenario [6] when the feature map is not encrypted. Table 1 lists the related methods and this work.

3 Gradient-Based Model Evasion Attacks Against Feature Extractor with Multiple Guide Images

In this section, we evaluated a gradient-based model evasion attack against feature extractor. Conventional works have assumed just one guide image, but we performed the attack using multiple guide images.

3.1 Experimental Setup

We used an image classification DNN model with VGG-16 architecture [15] trained by ImageNet [2]. The model had 13 convolution layers in the feature extractor and three fully connected layers in the classifier. The model was pretrained and published on Pytorch [11]. The top-1 accuracy of the model was 71.6%.

We extended the TMI-FGSM against feature extractor [8] to calculate adversarial examples by applying Eq. 4 instead of 3. The equation minimizes the distance between the intermediate feature map by adversarial examples I_{adv} and the mean of intermediate feature maps by guide images I_g, as

$$\begin{aligned} m^0 &= 0 \\ m^{k+1} &= m^k + \frac{\nabla_{I^k_{adv}} MSE(\phi(I^k_{adv}), \frac{1}{S}\sum_{s=1}^{S}\phi(I^s_g))}{||MSE(\phi(I^k_{adv}), \frac{1}{S}\sum_{s=1}^{S}\phi(I^s_g))||_1}, \end{aligned} \tag{4}$$

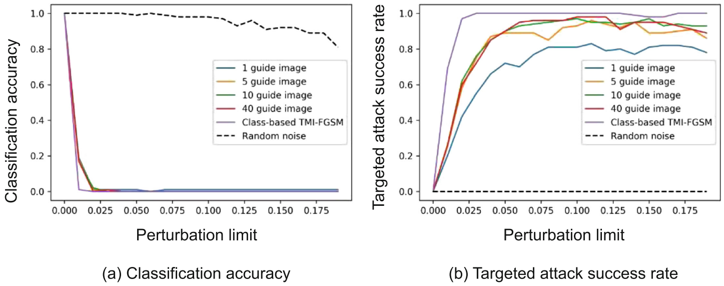
where S is the number of guide images and I^s_g is an s-th guide image.

We set 100 pairs of source images and multiple guide images. The brightness value of each image was normalized to the range of 0 to 1. We acquired a feature map from the 13th convolution layer output of the model. We evaluated the classification accuracy and the success rate of the targeted attack while increasing perturbation limits from ±0.01 to ±0.2. The classification accuracy is a ratio of the created adversarial examples that are classified into the correct (source) class. The success rate is a ratio of the created adversarial examples that are classified into the guide class. Note that not all samples behave as adversarial examples due to the limited amount of perturbation applied to the input image.

3.2 Experimental Results

Figure 10 shows the classification accuracy and the success rate of the targeted attack for each number of guide images. The results of the TMI-FGSM attack [3] and random noise attack which adds uniformed random noise to input images are also shown in the figures for comparison.

As shown in Fig. 10(a), the TMI-FGSM against feature extractor significantly reduced the classification accuracy, similar to using whole model. There was no difference in attack efficiency depending on the number of guide images. In (b), the TMI-FGSM against feature extractor with only one guide image achieved the targeted attack success rate of about 80%. The success rate of the attack



(a) Classification accuracy

(b) Targeted attack success rate

Fig. 10. (a) Classification accuracy and (b) success rate of targeted attack for each number of guide images.

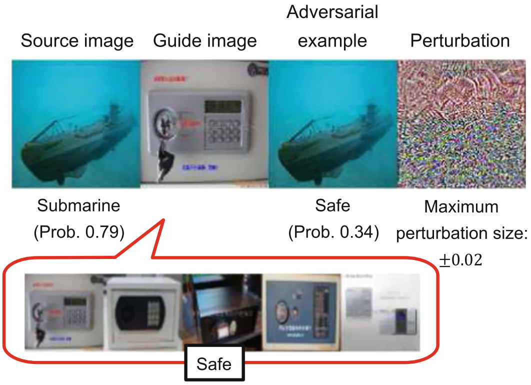


Fig. 11. Results of TMI-FGSM against feature extractor using five guide images.

was significantly improved when the number of images was increased from one to five. We improved the success rate to about 98% with 40 guide images. Thus, the success rate of the targeted attack was increased by increasing the number of guide images. However, a success rate comparable to the attack against the whole model was not achieved when the perturbation amount was low. There was almost no difference in attack efficiency depending on the number of guide images when there were five or more guide images.

Figure 11 shows an example of the TMI-FGSM against feature extractor using five guide images. The source image is from the submarine class and the guide image is from the safe class. The adversarial example was classified into the safe class. Figure 12 shows examples of the attack using one of these guide images. The targeting attack was not successful even if one of the guide images was used. However, the targeting attack was successful by averaging the feature maps of these five guide images, as shown in Fig. 11. This is a rare case but it indicates that an attack with multiple guide images can improve the success rate even if the attack with a single guide image has failed.

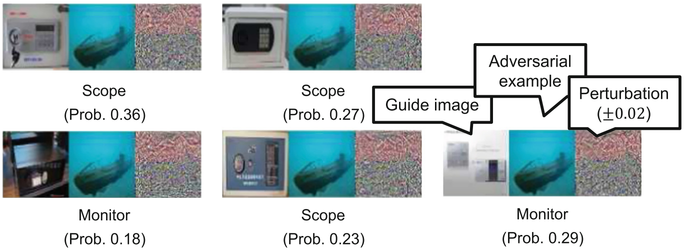


Fig. 12. Results of TMI-FGSM against feature extractor using one guide image.

4 Gradient-Based Model Evasion Attacks Against Feature Extractor in eNNclave Scenario

In this section, we performed a gradient-based model evasion attacks against feature extractor in eNNclave scenario [13]. We investigated how many layers could be placed in the normal world. When more layers are placed in the normal world, the device performs inference faster than when all layers are placed in the secure enclave.

4.1 Threat Model

An adversary attempts to create adversarial examples from source images against a target DNN model which is executed on a device. The device is designed with eNNclave scenario. The target DNN model is trained with transfer learning, and the feature extraction layer is the same as a public model's one. Users of the device, including the adversary, input a source image to the device and the device returns the classification result by the DNN model. Here, since the classification result returns only with the low-precision probability or the label name of the Top-1 class, the gradient estimation-based model evasion attacks with class probability cannot be performed [1,14]. The device calculates the feature extraction layer on the normal world application and inputs the result (intermediate feature map) to the enclave application. The enclave application calculates the classification layer and returns the modified classification result to the normal world application.

The goal of the adversary is to reduce the classification accuracy of the target model by adversarial examples and/or to classify an adversarial example into a certain guide class. The adversary can freely peep into the memory space managed by the normal world. When the DNN model inference is executed in the normal world, the adversary can read the DNN model structures and parameters. The adversary can not read or write to the memory space managed by the secure world (enclave).

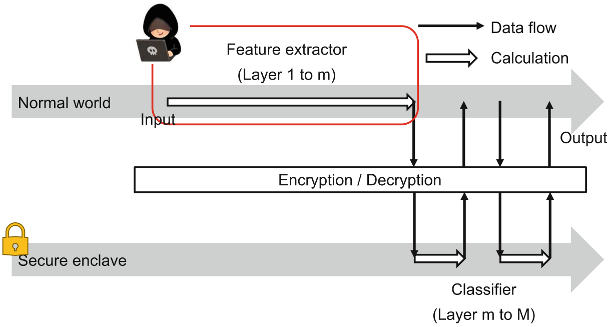


Fig. 13. Overview of our evaluation scenario based on eNNclave.

4.2 Experimental Setup

We assumed that the number of layers from the DNN model was M and that the layers from the first to m-th operated in the normal world while the remaining ones operated in the secure enclave. The adversary could steal the first to m-th layers as a feature extractor and perform an attack against it. Figure 13 shows an overview of our evaluation scenario based on eNNclave.

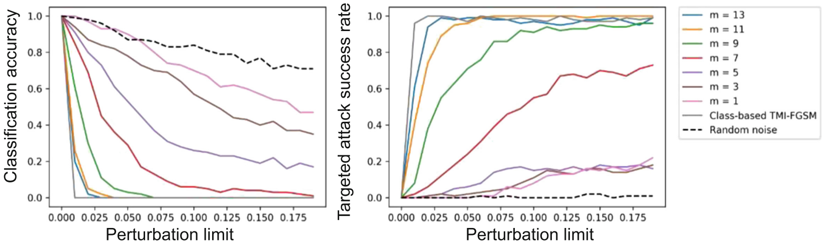
We trained a DNN model for a GTSRB traffic sign classification task [16] from the VGG-16 ImageNet classification model by using transfer learning. The model achieved a classification accuracy of 90.3%. We set 100 pairs of source images and ten guide images and then performed the attack. We evaluated the classification accuracy and the success rate of the targeted attack while increasing the perturbation limits from ±0.01 to ±0.2.

4.3 Experimental Results

Figure 14 shows the classification accuracy and the success rate of the targeted attack for each number of guide images. The results of TMI-FGSM against the whole model and random noise attack which adds uniformed random noise to input images are also shown in the figures for comparison. A result of the TMI-FGSM attack against feature extractor (m = 13) is shown in Fig. 15.

As we can see in Fig. 14(a), it was difficult to decrease the classification accuracy of the model when the adversary was able to use only shallow layers. However, the attack degraded the accuracy more efficiently than random noise even if the adversary knew only the first convolution layer (i.e., m was one). In (b), it was easier for the adversary to fool the model when more layers were executed in the normal world (i.e., m was more than nine).

The adversary has access to the model parameters if the calculations are carried out in the normal world. These experimental results show that the adversary can perform a gradient-based model evasion attack against the feature extractor even if it only knows the first several layers of the model parameters.



(a) Classification accuracy

(b) Targeted attack success rate

Fig. 14. (a) Classification accuracy and (b) success rate of targeted attack for each adversary's accessible layers m.

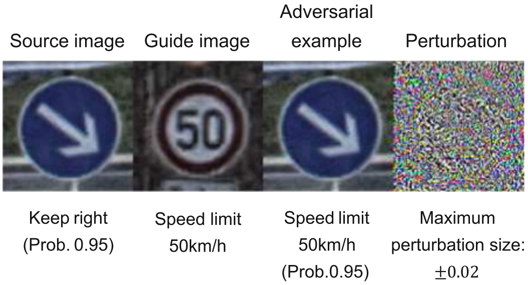


Fig. 15. Results of TMI-FGSM against feature extractor (m = 13) using 10 guide images.

5 Gradient Estimation-Based Model Evasion Attacks with Feature Maps

In this section, we evaluated a gradient estimation-based model evasion attacks with feature maps. We assumed that intermediate feature maps were stored in the normal world memory with plain text while the secure enclave switched the calculation layer. An adversary could attack by using a feature map from each calculation layer output.

5.1 Threat Model

(9) An adversary attempts to create adversarial examples from source images against a target DNN model which is executed on a device. The device is designed with MLCapsule scenario. Users of the device, including the adversary, input a source image to the device and the device returns the classification result by the DNN model. (9, 10, 12) Here, since the classification result returns only with the low-precision probability or the label name of the Top-1 class, the gradient estimation-based model evasion attacks with class probability cannot be performed [1,14]. (9) The device calculates the DNN model inference process layer-by-layer on the enclave application. The enclave application calculates a layer and

Algorithm 2. Extended SimBA for exploiting feature map. Parts that differ from SimBA are highlighted in red.

```
Input: Source image $I_s$, Guide images $I_g$, Number of guide images $S$, Number of
  image pixels $N$,
  Perturbation parameter $\alpha$, DNN model $f$, Feature extractor $\phi$
Output: Adversarial example $I_{adv}$
  $I_{adv} = I_s$
  for n=0 to N-1 do
    $x' = I_{adv}$
    $x'[n] = I_{adv}[n] + \alpha$
    $p^+ = MSE(\phi(x'), \frac{1}{S}\sum_{s=1}^{S}\phi(I_g^s)))$
    $x' = I_{adv}$
    $x'[n] = I_{adv}[n] - \alpha$
    $p^- = MSE(\phi(x'), \frac{1}{S}\sum_{s=1}^{S}\phi(I_g^s)))$
    if $p^+ < p^-$ then
      $I_{adv}[n] = I_{adv}[n] + \alpha$
    else if $p^+ > p^-$ then
      $I_{adv}[n] = I_{adv}[n] - \alpha$
    else
      $I_{adv}[n] = I_{adv}[n] + 0$
    end if
    $I_{adv} = clip(I_{adv})$
  end for
  return $I_{adv}$
```

temporarily stores intermediate feature maps into normal world memory space for preparing the next layer. Each intermediate feature map is not encrypted.

The goal of the adversary is to reduce the classification accuracy of the target model by adversarial examples and/or to classify an adversarial example into a certain guide class. The adversary can freely peep into the memory space managed by the normal world. When the intermediate feature map is stored in the normal world memory space, the adversary can read the feature map. The adversary can not read or write to the memory space managed by the secure world (enclave). Thus, the adversary can not obtain the DNN model structure and parameters, the adversary only exploits the feature map.

5.2 Attack Methodology

Algorithm 2 shows our gradient estimation-based model evasion attack with feature maps. It is an extended method from class probability-based SimBA to exploiting feature maps. An adversary inputs source image I_s and some guide images I_g and obtains intermediate feature maps $\phi(I_s)$ and $\phi(I_g)$. The adversary changes the pixels of the source image in the plus and minus directions, respectively, and measures the distance between $\phi(I_s)$ and $\phi(I_g)$. The adversary estimates a gradient of a source image with respect to the distance and selects the perturbation of the pixel.

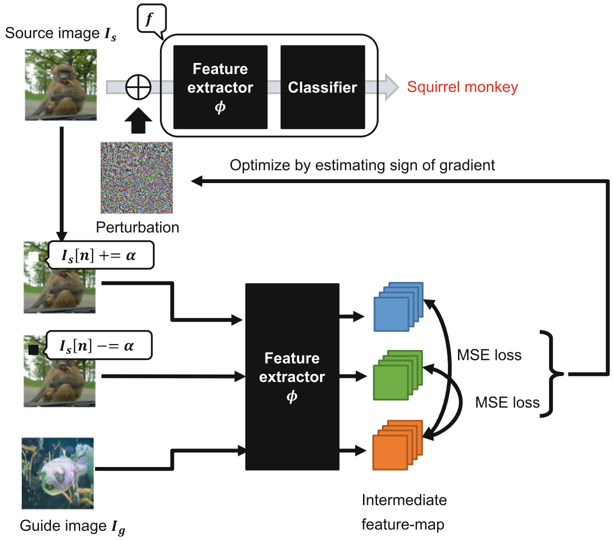


Fig. 16. Overview of gradient estimation-based model evasion attack with feature maps

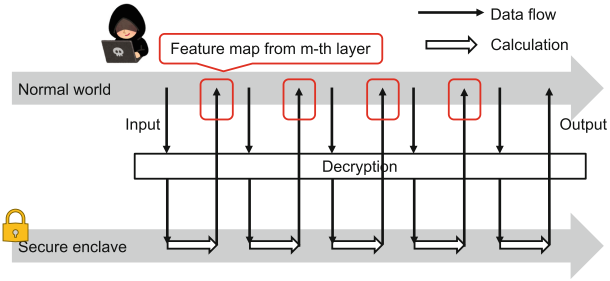
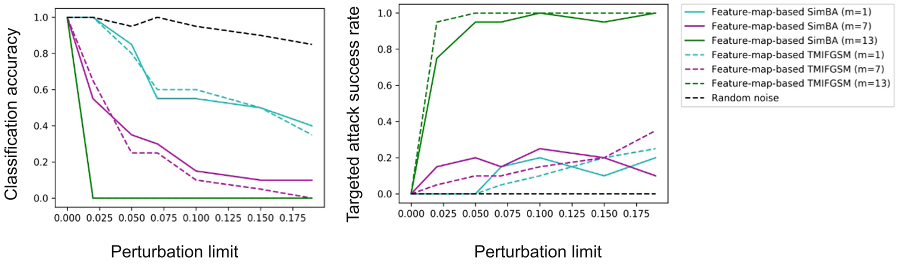


Fig. 17. Overview of our gradient estimation-based model evasion attack scenario based on MLCapsule.

Figure 16 shows an overview of the gradient estimation-based model evasion attack with feature maps. The adversary can not obtain a feature extractor but can obtain an intermediate feature map from each calculation layer. The adversary chooses the perturbation of each pixel on the basis of the MSE loss between the intermediate feature map of the input image with tampered pixel and guide images (Fig. 17).

5.3 Experimental Setup

We prepared the same DNN model for traffic sign recognition as Sect. 4.2. We assumed an adversary could obtain intermediate feature maps from each output of the 1st, 7th, and 13th layer. We set 20 pairs of source images and 10 guide



(a) Classification accuracy

(b) Targeted attack success rate

Fig. 18. (a) Classification accuracy and (b) success rate of targeted attack by SimBA with feature map, TMI-FGSM against feature extractor, and random noise.

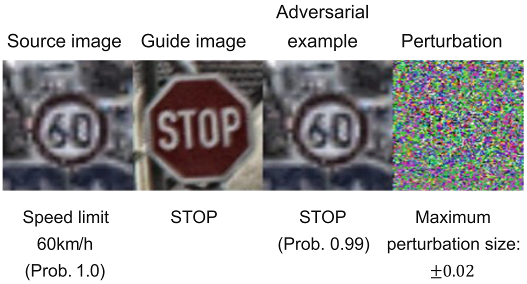


Fig. 19. Results of SimBA with feature map (m = 13) using 10 guide images.

images. We evaluated the success rate of the targeted attack and classification accuracy while increasing the perturbation limits from ±0.01 to ±0.2.

5.4 Experimental Results

Figure 18 shows the classification accuracy and the success rate of the targeted attack and by using intermediate feature maps from the feature extractor. The results of the TMI-FGSM against feature extractor and random noise attack, which adds uniformed random noise to input images, are also shown for comparison. A result of the SimBA with feature map (m = 13) is shown in Fig. 19.

As shown in Fig. 18(a), the SimBA with feature map significantly reduced the classification accuracy, similar to the TMI-FGSM against feature extractor. In (b), it also achieved a targeted attack success rate comparable to the TMI-FGSM against feature extractor. These results indicate that the gradient estimation-based model evasion attack with feature map is a threat to TEEs and it is necessary to encrypt feature maps in the normal world.

It requires additional execution time to encrypt and store the intermediate feature map in the normal world memory, but it is more important to prevent a gradient estimation-based model evasion attack with feature map.

6 Conclusion

In this work, we investigated which information of DNN model inference should be concealed from an adversary to prevent model evasion attacks. We assumed a DNN inference process performed in an isolated execution environment where the encrypted DNN model is decrypted and processed.

First, we improved the TMI-FGSM against feature extractor by using multiple guide images. Conventional techniques have used one guide image and achieved a targeted attack success rate of about 80%. We used five or more guide images and achieved a success rate of about 98% with 40 guide images.

Second, we performed the TMI-FGSM against feature extractor in an eNNclave scenario [13] and evaluated how many layers could be placed in the normal world. It was easier for an adversary to fool the model when more layers were executed in the normal world. The adversary fooled the model more efficiently than random noise even if it knew only the shallowest layer. These results demonstrate that an adversary can perform a gradient-based model extraction attack against feature extractor if even a part of the inference operations is calculated in the normal world.

Finally, we evaluated a gradient estimation-based model evasion attack with feature maps. We assumed that intermediate feature maps were stored in the normal world memory with plain text while the secure enclave switched the calculation layer. Our extended SimBA significantly reduced the classification accuracy, similar to the TMI-FGSM against the feature extractor. It also achieved a targeted attack success rate comparable to the TMI-FGSM. Additional execution time is required to encrypt and store the intermediate feature map in the normal world memory, but it is important to prevent a gradient estimation-based model evasion attack with feature maps.

Our findings demonstrate that DNN model parameters and intermediate feature maps should be concealed not only for protecting intellectual property but also for preventing model evasion attacks. To ensure safe and fast DNN inference, an isolated execution environment that allows accessing large memory space and DNN inference accelerators from a secure enclave is required.

Acknowledgments. This work was supported by JST-Mirai Program Grant Number JPMJMI19B6, Japan.

References

1. Bhagoji, A.N., He, W., Li, B., Song, D.: Practical black-box attacks on deep neural networks using efficient query mechanisms. In: Proceedings of the European Conference on Computer Vision (ECCV), September 2018
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database, pp. 248–255. Institute of Electrical and Electronics Engineers (IEEE), March 2010
3. Dong, Y., et al.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 9185–9193, October 2017

4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR, December 2015
5. Graepel, T., Lauter, K., Naehrig, M.: ML confidential: machine learning on encrypted data. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 1–21. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37682-5_1
6. Hanzlik, L., et al.: MLCapsule: guarded offline deployment of machine learning as a service, August 2018
7. Huang, Q., Katsman, I., He, H., Gu, Z., Belongie, S., Lim, S.N.: Enhancing adversarial example transferability with an intermediate level attack. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4732–4741, July 2019
8. Inkawhich, N., Wen, W., Li, H.H., Chen, Y.: Feature space perturbations yield more transferable adversarial examples. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2019-June, pp. 7059–7067. IEEE Computer Society, June 2019
9. Mohassel, P., Zhang, Y.: SecureML: a system for scalable privacy-preserving machine learning. In: Proceedings - IEEE Symposium on Security and Privacy, pp. 19–38. Institute of Electrical and Electronics Engineers Inc., June 2017
10. Narodytska, N., Kasiviswanathan, S.: Simple black-box adversarial attacks on deep neural networks. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol. 2017-July, pp. 1310–1318, August 2017
11. Paszke, A., et al.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv, December 2019
12. Sabour, S., Cao, Y., Faghri, F., Fleet, D.J.: Adversarial manipulation of deep representations. In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, November 2015
13. Schlögl, A., Böhme, R.: eNNclave: offline inference with model confidentiality. In: Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security. ACM, New York (2020)
14. Senzaki, Y., Ohata, S., Matsuura, K.: Simple black-box adversarial examples generation with very few queries. IEICE Trans. Inf. Syst. **103**(2), 212–221 (2020)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, September 2015
16. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. **32**, 323–332 (2012)
17. Wang, B., et al.: With great training comes great vulnerability: practical attacks against transfer learning. In: USENIX, pp. 1281–1297 (2018)
18. Xie, P., Bilenko, M., Finley, T., Gilad-Bachrach, R., Lauter, K., Naehrig, M.: Crypto-Nets: Neural Networks over Encrypted Data, December 2014