# Adversarial Detection with Gaussian Process Regression-based Detector

**Sangheon Lee[1], Noo-ri Kim[1], Youngwha Cho[1], Jae-Young Choi[1], Suntae Kim[2], Jeong-Ah Kim[3] and Jee-Hyong Lee[1*]**

[1] College of Information and Communication Engineering, Sungkyunkwan University
Suwon 16419, South Korea
[e-mail: {lawlee1, pd99j, choyh2285, jaeychoi, john}@skku.edu]
[2] Department of Software Engineering, Chonbuk National University
Jeonju 54896, South Korea
[e-mail: stkim@jbnu.ac.kr]
[3] Department of Computer Education, Catholic Kwandong University
Gangneung 25601, South Korea
[e-mail: clara@cku.ac.kr]
*Corresponding author: Jee-Hyong Lee

*Received March 4, 2019; revised May 3, 2019; revised June 11, 2019; accepted August 3, 2019; published August 31, 2019*

## *Abstract*

Adversarial attack is a technique that causes a malfunction of classification models by adding noise that cannot be distinguished by humans, which poses a threat to a deep learning model. In this paper, we propose an efficient method to detect adversarial images using Gaussian process regression. Existing deep learning-based adversarial detection methods require numerous adversarial images for their training. The proposed method overcomes this problem by performing classification based on the statistical features of adversarial images and clean images that are extracted by Gaussian process regression with a small number of images. This technique can determine whether the input image is an adversarial image by applying Gaussian process regression based on the intermediate output value of the classification model. Experimental results show that the proposed method achieves higher detection performance than the other deep learning-based adversarial detection methods for powerful attacks. In particular, the Gaussian process regression-based detector shows better detection performance than the baseline models for most attacks in the case with fewer adversarial examples.

***Keywords:*** Adversarial Attack, Adversarial Defense, Adversarial Detection, Gaussian Process Regression, Image Classification

A preliminary version of this paper was presented at ICONI 2018, and was selected as an outstanding paper.
This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C4A7030503). Also, this research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069440).

**http://doi.org/10.3837/tiis.2019.08.027** **ISSN : 1976-7277**

## 1. Introduction

In recent years, the development of machine learning, especially deep learning, and its applications have been actively studied [1]-[5]. The deep learning model has been applied to various fields, such as image classification [6]-[10], natural language processing [11]-[14], semantic analysis [15][16], and object detection [17][18] and shows high performance as a state-of-the-art technique. Although deep learning is applied in many areas, the adversarial attacks that have recently been proposed raise questions about the reliability of the deep learning model.

The adversarial attack is a technique that changes the results of a classification or regression model by mixing perturbations that are imperceptible to human in the input data of the model. Recently, adversarial attack methods that deceive the image classification neural network model have been actively studied [19]-[23]. For a given natural image x that has no perturbation, the adversarial attack produces an image x' that is visually similar, but has a different classification result. The x' is called an adversarial example. By creating adversarial examples through the attack, attackers can mislead the neural network model. **Fig. 1** at right shows an image in which the adversarial attack is applied to the image at left recognized as "panda" by the image classification model. The two images are not distinguished by the human eye, but the image classification model recognizes the right image as "gibbon", rather than "panda".

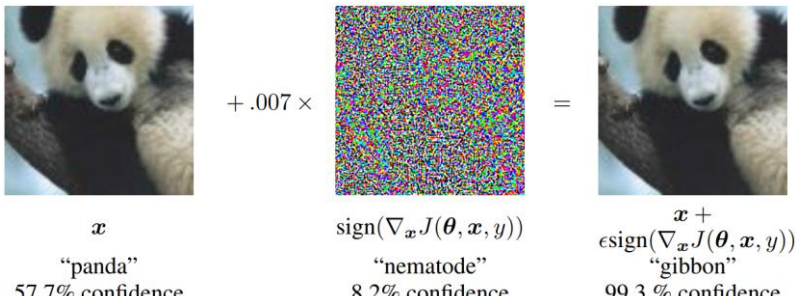


**Fig. 1.** An adversarial image that has imperceptible perturbation added to the natural image [19].

If a deep learning model is applied to a major part of the system, the adversarial attack can lead to serious problems in system security. For example, when an adversarial attack is applied to the deep learning model that is part of an autonomous vehicle, the model interprets the current scene differently, and a fatal accident may occur. To prevent this, several methods have recently been proposed to protect against adversarial attack [24][25]. Adversarial defense aims to get the result of x for a given adversarial example x'. Many adversarial defense methods have been proposed, such as increasing the robustness of the model by augmenting the training data of the classification model [19], or creating a more robust network by distilling the original classification model [24]. However, few defense methods that can effectively defend against various and powerful attacks have yet been proposed.

Other researchers proposed adversarial detection instead of adversarial defense, which determine whether a given image is an adversarial example or not [26]-[29]. If the detector determines that the given image is an adversarial example, the image classification model can prevent the attack, by rejecting the classification task of the adversarial image. From the viewpoint of the service system with a deep learning model, it is only necessary to reject the input causing the malfunction of the system, without having to give an accurate result for that.

Adversarial detection can initially block the provision of information about the decision boundary of the deep learning model to the attacker. Therefore, the attacker may receive limited information about the model, making it more difficult for the attacker to perform a more sophisticated attack [30].

However, many of the detection methods already proposed are the deep neural network-structured model, which requires a large number of adversarial examples to train. From the point of view of the system to which the deep learning model is applied, it is more effective to perform detection with only a few adversarial examples, since adversarial examples of a number of attacks can be secured. In particular, applying an attack that generates adversarial examples using gradients of the model can generate the adversarial example that deceives not only the classifier model but also the detector, and the detection method applied to the model can be useless [30]. Therefore, adversarial detection to effectively prevent adversarial attack requires the following characteristics:

1. Detection should be performed with high performance, even with a small number of adversarial examples.
2. Adversarial detection methods should be non-differentiable, so that gradient-based attacks cannot be applied to detection methods.

In this paper, we propose an efficient detection method for adversarial example. The proposed method consists of two steps. First, the intermediate feature values generated by the classification model are extracted for a given image. Second, the intermediate feature information is used to determine whether the image is an adversarial example by applying Gaussian process regression [31]. Gaussian process regression measures the correlation information of given data into a covariance matrix, and performs regression based on it. Therefore, classification for detecting adversarial examples can be performed effectively with only a small amount of given data. In addition, the output function $f(x)$ for the input $x$ is non-differentiable, because the Gaussian process regression trains the probabilistic distribution of the output using the observed data. Therefore, the secondary adversarial attack on Gaussian process regression does not work. Experimental results show that our detection model has good results with fewer adversarial examples than other neural network-based detectors. In particular, for powerful attacks with high attack success rates, our detection model outperforms the baseline model. In addition, the proposed model showed better detection performance than the baseline model for the whole attacks, in the case where the learning data is extremely small.

Section 2 describes the adversarial attack, adversarial detection, and the Gaussian process regression used in the proposed method. Section 3 describes the proposed method, while Section 4 presents experiments and results to verify the performance of our detection method. Section 5 concludes the paper.

## 2. Related Work

### 2.1 Adversarial Attack

The basic purpose of the adversarial attack is to create an example with a minimal perturbation that looks similar to a natural image, but causes the target model to be misclassified. The adversarial attack has been actively studied, especially for deep learning models that perform

image classification. For a given deep learning model $f$, an adversarial example $x'$ for the natural image $x$ is generated by the following constrained optimization problem [32]:

$$\begin{aligned} &\text{argmin}_{x'} \|x' - x\| \\ &\text{s.t.} \quad f(x') \neq f(x), \\ &\qquad x' \in [0,1] \end{aligned} \tag{1}$$

where, $f(x')$ and $f(x)$ denote output classes of the model for $x'$ and $x$, respectively, and $\|\cdot\|$ denotes the distance between two images. That is, the adversarial attack is an optimization problem that minimizes the size of the perturbation $\|x' - x\|$ mixed to the natural image $x$, under the condition that the classification result by the model is different from the natural image. From the viewpoint of the decision boundary of the model, the adversarial example is a data point that belongs to a different class from the natural image, but that is located very close to the decision boundary of the natural class. This is because although the adversarial example is located at a very small perturbation distance from the natural image, the output class by the model is different [33].

Recently, various types of adversarial attack have been proposed by researchers. Goodfellow et al. [19] introduced the Fast Gradient Sign Method (FGSM), an attack that uses the gradient value of the loss function of a model for a given natural image. The FGSM is expressed as follows:

$$x' = x + \epsilon \text{sign}\left(\nabla_x J(x, y_{true})\right) \tag{2}$$

where, $J(\cdot,\cdot)$ denotes the loss function of the model (e.g. cross-entropy), and $\epsilon$ is epsilon, which indicates the size of the perturbation mixed in the natural image. Since the gradient of the model contains the direction information to the decision boundary of the true class for a given image, FGSM creates an adversarial example by adding a perturbation of magnitude ε to the image in the opposite direction of the gradient generated by the model. **Fig. 1** shows the image generated by the FGSM.

Kurakin et al. [20] proposed an iterative version of the FGSM, the Basic Iterative Method (BIM). BIM is an attack that generates adversarial examples by repeatedly applying FGSM as small steps, and is also called Iterative FGSM (I-FGSM). BIM is expressed as follows:

$$x'_0 = x, \;\; x'_{n+1} = Clip_{x,\epsilon}\{x'_n + \alpha \,\text{sign}\left(\nabla_x J(x'_n, y_{true})\right)\} \tag{3}$$

BIM carries out a search on the assumption that an adversarial example exists in the e-neighborhood based on the natural image. BIM can also perform a targeted adversarial attack that causes the output class of the adversarial example to be a specific target class. This method is an Iterative Target Class Method, expressed by the following equation:

$$x'_0 = x, \;\; x'_{n+1} = Clip_{x,\epsilon}\left\{x'_n - \alpha \,\text{sign}\left(\nabla_x J\left(x'_n, y_{target}\right)\right)\right\} \tag{4}$$

If BIM is performed during sufficient iterations, it is observed that the adversarial example generated by this attack can always have a target class as an output by the model.

Papernot's Jacobian-based Saliency Map Attack [21] performs targeted attacks through the adversarial saliency map. The saliency map represents the influence of each input feature on

the output of the model for a given image, and is computed through a Jacobian matrix, as follows:

$$J_F(x) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial F_j(x)}{\partial x_i}\right]_{i \times j} \tag{5}$$

Based on the saliency map, JSMA generates adversarial examples by changing only those features with a high saliency map value in the natural image, that is, those having a large influence on output determination. The model can easily be deceived with only a small perturbation generated through JSMA. Experimental results show that the attack rate of 97 % is achieved by modifying only 4.02 % of the input feature. However, it has the disadvantage that the attack time is long, because of the high computational cost in updating the saliency map.

Moosavi-Dezfooli et al. [22] proposed the DeepFool attack that updates the perturbation vector for the natural image every iteration, and performs the algorithm, until the result image is misclassified for the first time. The DeepFool attack generates an adversarial example by adding noise iteratively using a perturbation vector with the direction of the nearest decision boundary from the given image, based on the assumption that an adversarial example exists near the decision boundary of the model.

Carlini and Wagner [23] proposed the C&W attack that is a kind of targeted attack, and has better performance than the other attacks proposed so far. They set the loss function that is low on the adversarial example and high on the natural image, and perform an adversarial attack by minimizing it. The strongest L2 loss among the losses they searched for is [34]:

$$\min_{x'} \|x' - x\|_p + c \max_{i \neq y_{target}} \left(f(x')_i - f(x')_{y_{target}}, -k\right) \tag{6}$$

where, the adversarial example $x' = \frac{1}{2}(\tanh(w) + 1)$, and $c$ and $k$ are parameters.

In addition to the attacks mentioned above, various attack methods have been proposed that modify existing attacks, or have new methods. The proposed adversarial attacks are applicable to different models in various data, and show high attack success rates. The question about the reliability of the deep learning model is raised through adversarial attack studies. Since a fundamentally robust deep learning model for such adversarial attacks has not yet been proposed, an additional method that can defend attacks is needed, such as adversarial defense or adversarial detection.

## 2.2 Adversarial Detection

Powerful adversarial attacks with various algorithms have been proposed, and thus in order to defend against such attacks, an adversarial defense has been proposed. The basic purpose of the adversarial defense is to allow a given model to produce a true label for the adversarial example input. Goodfellow et al. [19] proposed a typical defense method called adversarial training. It is a kind of data augmentation that adds adversarial examples to the training data of the model. Adversarial training successfully defended against the adversarial attack method that used the augmentation, but failed to effectively defend against other attacks. In particular, adversarial training cannot effectively defend against the secondary attack that attacks the model with the defense method as a new target network. Adversarial training, as well as various adversarial defense methods, have been proposed, but most of them have shown high

performance only for specific attacks, and it is impossible to defend against the newly proposed, powerful attacks.

Recently, adversarial detection has been studied to prevent adversarial attack by a rather simple method, instead of adversarial defense. Adversarial detection is a technique for judging whether a given image is an adversarial example, and has been recently studied with adversarial defense. From the perspective of a system with a deep learning model, it is possible to further secure the system by rejecting the adversarial example input, after determining through adversarial detection whether the given input is an adversarial example. Because the adversarial example is a data point located near the decision boundary of the model, rejecting the result for the adversarial example can minimize the propagation of information about the criteria that the model determines the output for a given input to the attacker. This makes it more difficult for an attacker to perform more sophisticated attacks that can deceive the target model with higher probability [30].

Various adversarial detection methods have been proposed by many researchers. Grosse et al. [26] proposed a modification of adversarial retraining to detect the adversarial example. For a classification model with $N$ result classes, a new $N+1$ th class corresponding to the adversarial image is added to perform detection, and the model is trained using natural images and adversarial images. If the existing training dataset of the model is $(x_i, y_i) \in X_{origin}$, the new training set of the model $X_{new}$ is as follows:

$$X_{new} = X_{origin} \cup \{(x_i', N+1)\} \tag{7}$$

where, $x_i'$ denotes the adversarial example generated by applying a specific attack to the model trained with $X_{origin}$. Adversarial retraining showed good detection performance for the MNIST dataset but showed poor detection performance of 70 % detection rate and 40 % false positive rate for the CIFAR10 dataset [30].

Metzen et al. [27] proposed a detector that performs detection using the output value of the inner convolution layer of the classification model. **Fig. 2** shows that the proposed detector is a deep learning model that consists of convolution layers and max pooling layers. Metzen's detection method showed high detection accuracy in experiments using the MNIST and CIFAR10 datasets.

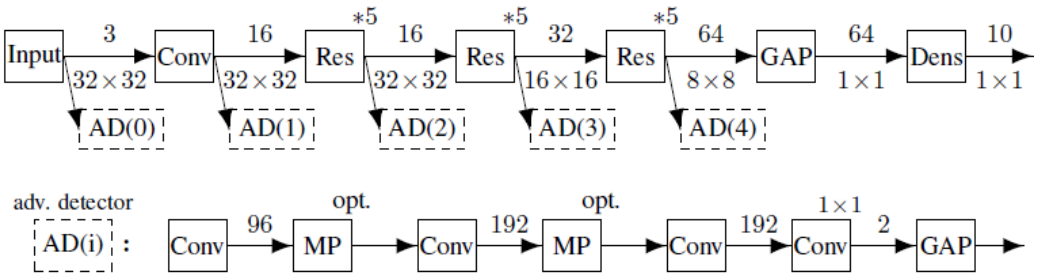


**Fig. 2.** The deep learning structured adversarial detector proposed by Metzen et al. [27].

Several adversarial detection methods have been proposed that use deep neural networks structured detectors, such as the detector proposed by Metzen, and most of them show high detection performance [28][29]. However, such a deep neural network-based detection method has the disadvantage that it requires a large number of adversarial examples to train the detector. From the perspective of a system with a deep learning model, the number of

adversarial examples acquired is equal to the number of attacks that the system received. In other words, the system is able to defend the attack with high performance through a deep neural network-based detector only after obtaining enough information about the attack by receiving a large number of attacks. Therefore, the deep learning-based adversarial detection method cannot be said to effectively prevent the adversarial attack.

### 2.3 Gaussian Process Regression

Gaussian process is a random process in which every finite collection of random variables has a multivariate normal distribution. Gaussian process regression is a technique to infer the mean and variance of the whole data range based on the observed data, by defining the relationship between the data using the characteristics of the data, assuming the distribution of the data follows the Gaussian process [31][35]-[37]. Assuming $f(x)\sim N\left(0, K(\theta, x, x')\right)$ for the function $f(x)$ of $x$, the log marginal likelihood is as follows:

$$\log p(f(x)|\theta, x) = -\frac{1}{2} f(x)^T K(\theta, x, x')^{-1} f(x) \; -\frac{1}{2}\log\det\left(K(\theta, x, x')\right) \; -\frac{|x|}{2}\log 2\pi \quad (8)$$

where, $K(\theta, x, x')$ is a covariance matrix for all possible observed data pairs $(x, x')$, calculated from a pre-defined kernel function, and $\theta$ is a hyperparameter of the covariance function. Based on the $\theta$ that maximizes this marginal likelihood, the distribution of the function value $f(x^*)$ for the unobserved data $x^*$is $p(y^*|x^*, f(x), x) = N(y^*|A, B)$. That is, the posterior distribution has mean function $A$ and variance function $B$, where $A$ and $B$ are calculated through the following equations:

$$\begin{aligned} A &= K(\theta, x^*, x)K(\theta, x, x')^{-1} f(x) \\ B &= K(\theta, x^*, x^*) - K(\theta, x^*, x)K(\theta, x, x')^{-1} K(\theta, x^*, x)^T \end{aligned} \quad (9)$$

where, $K(\theta, x^*, x)$ denotes the covariance values between all observed data $x$ and the new data $x^*$ based on the hyperparameter value $\theta$, and $K(\theta, x^*, x^*)$ is the variance value at $x^*$. **Fig. 3** shows the distribution of the function obtained by the Gaussian process regression as mean function and variance function. Consequently, the value of the function for the unobserved data $x^*$ can be predicted through the mean function, and the variance function implies the uncertainty of the function.

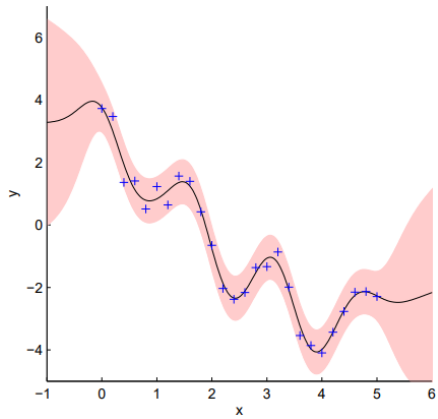


**Fig. 3.** Distribution of functions calculated by Gaussian process regression. The shaded area represents the 95 % confidence interval [31].

Since Gaussian process regression defines prior and predicts posterior in consideration of covariance between data, it is possible to obtain more accurate regression results with only a small number of data for the data that follow the Gaussian process, than by a general regression method. Our proposed detection method works based on Gaussian process regression, so that it can achieve high detection accuracy with only a small number of adversarial examples.

## 3. Gaussian Process Regression-based Detector

We define the characteristics of the adversarial detection method as follows to effectively prevent the adversarial attack. First, the adversarial detection method should operate with high detection accuracy with only a small number of adversarial examples. This is because the number of adversarial examples held by the system is equal to the number of attacks received. It is also directly associated with the shortcomings of existing deep learning-based detectors. Second, the adversarial detection method should be non-differentiable, or it must be difficult for the attacker to obtain the gradient of the detector. This is to prevent secondary attacks that use the gradient of the detector to generate an adversarial example that can fool not only the target model, but also the detector.

In order to satisfy these properties, we propose a method for detecting adversarial examples based on the Gaussian process regression. First, we extract the intermediate features generated by the pre-trained classification model for natural or adversarial images. The intermediate feature is the output vector of the model's last hidden layer, whose dimension is the class number of the image set. If the natural image set $\{x_i\}$ and the adversarial image generated by applying adversarial attack to each xi is $\{x_i'\}$, the extracted intermediate feature set $X_{inter}$ is as follows:

$$X_{inter} = \{f(x) \mid x \in \{x_i\} \cup \{x_i'\}\} \tag{10}$$

Second, we use the extracted intermediate feature set as the observed data of the Gaussian process regression-based detector. The observed dataset for fitting the Gaussian process regression-based detector is as follows:

$$D_{observed} = \{(x_{inter}^i, y^i)\} \tag{11}$$
$$s.t. \quad x_{inter}^i \in X_{inter}$$
$$y^i = \begin{cases} 0 & if\ x_{inter}^i\ is\ natural\ image \\ 1 & if\ x_{inter}^i\ is\ adversarial\ image \end{cases}$$

In the application process for the real model, the intermediate feature is extracted through a pre-learned model for a given input image, and the result of the Gaussian process regression is obtained when the extracted value is input.

**Fig. 4** shows the structure of our proposed adversarial image detector. The output value of the model's last hidden layer is the classification probability value for the image. In the case of adversarial images far from the two centers of the image classification boundary, classification

probability values for the two classes tend to be similar to each other. The detector would train this information to perform detection.

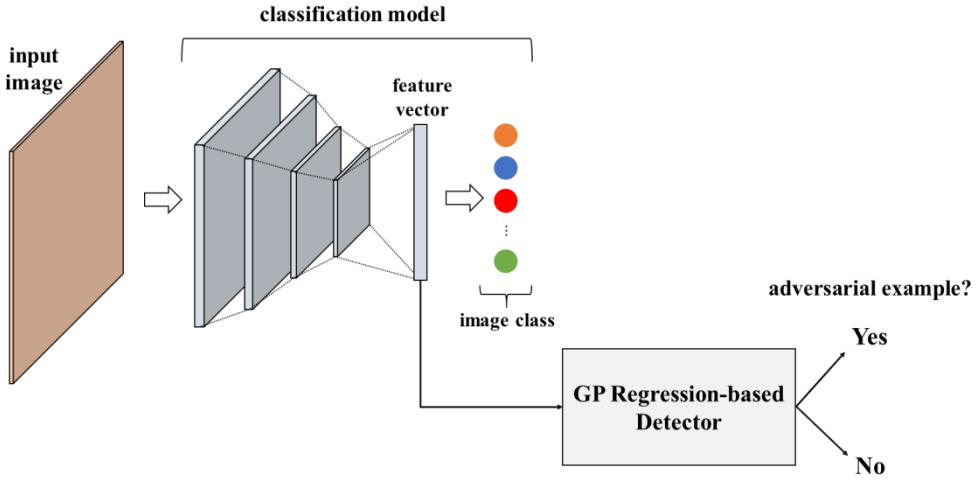


**Fig. 4.** Gaussian process regression-based adversarial image detector.

In the Gaussian process regression, the influence between two similar data is defined as covariance. If the dimension of data is high, it is difficult to grasp the pattern of covariance between data. Therefore, inputting low-dimensional high-level features extracted through convolution and pooling layers, rather than a high-dimensional raw image, might perform the Gaussian process regression more efficiently.

## 4. Experiments

To verify the performance of our proposed Gaussian process regression-based detector, the datasets used in the experiments are MNIST and CIFAR10. Sections 4.1 and 4.2 describe the classification models for datasets that are deep convolutional neural network-structured models, while Section 2 described the attack methods used in the experiments, which are the FGSM, BIM, JSMA, DeepFool, and C&W attacks.

**Table 1** shows the accuracy of the classification model for adversarial images, and the average L2 distance of perturbations generated by each attack. According to the definition of the adversarial attack, the smaller the perturbation generated by the attack and the lower the classification accuracy for the target model, the more powerful the attack. As a result of applying the attack against datasets and classification models used in the experiments, the C&W attack produced the least perturbation, and generally showed low classification accuracy. Therefore, the C&W attack is the most powerful attack among the five adversarial attacks used in the experiments.

For the Gaussian process regression-based detector, 300 natural images and 300 adversarial examples are used for training. The covariance function used in the proposed detector is the squared exponential function [31], which is as follows:

$$K_{SE}(x, x') = \exp\left(-\frac{|d^2|}{2\ell^2}\right) \tag{12}$$

The baseline model compared with our detector is the deep convolutional neural network-structured binary classification model proposed by Gong et al. [29], and the training data of the baseline model is set to 300 natural images and 300 adversarial examples for the same experimental conditions.

**Table 1.** Classification model accuracy for adversarial images and average L2 distance of perturbations.

| | FGSM | | BIM | | JSMA | | DeepFool | | C&W | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L2 | Acc. (%) | L2 | Acc. (%) | L2 | Acc. (%) | L2 | Acc. (%) | L2 | Acc. (%) |
| MNIST | 6.47 | 8.20 | 5.66 | 0.60 | 5.11 | 0.13 | 1.86 | 0.63 | 1.43 | 0.63 |
| CIFAR10 | 1.94 | 14.41 | 0.97 | 5.85 | 3.88 | 1.03 | 0.11 | 5.95 | 0.08 | 1.07 |

**Table 2.** Detection accuracy for the MNIST dataset.

| | FGSM (%) | BIM (%) | JSMA (%) | DeepFool (%) | C&W (%) |
|---|---|---|---|---|---|
| Baseline | **99.61** | **99.27** | 82.99 | 66.55 | 61.58 |
| GP-based | 92.86 | 69.3 | **97.94** | **99.64** | **99.67** |

**Table 3.** Detection accuracy for the CIFAR10 dataset.

| | FGSM (%) | BIM (%) | JSMA (%) | DeepFool (%) | C&W (%) |
|---|---|---|---|---|---|
| Baseline | 62.22 | 50.13 | **95.81** | 50.00 | 50.01 |
| GP-based | **76.92** | **50.42** | 94.86 | **97.94** | **97.93** |

### 4.1 MNIST dataset

The model for classifying the MNIST dataset is a simple 5-layer convolution neural network consisting of two convolution layers, one max pooling layer, and two dense layers. For training, 60,000 of 28×28×1 MNIST images are used, while for validation, 10,000 images are used. The optimizer used for model training is Adadelta [38], the training epoch is 20, learning rate is 0.001, and batch size is 128. As a result of the training, the accuracy of the classification model for the MNIST data is 99.3 %. The hyperparameters of the five attacks are set as follows; for FGSM and BIM, $\varepsilon$ is 0.4. For the C&W attack, we set the maximum iterations to 1,000, the initial constant to 0.001, and the learning rate to 0.005.

**Table 2** shows the experimental results for the MNIST dataset. For the FGSM and BIM attacks, the detection accuracies were relatively lower than the baseline detection model, but for the DeepFool, JSMA, and C & W attacks, which have higher attack success rates, our model is far superior to the baseline model. Since the detection accuracies of the baseline model for the DeepFool and C&W attacks are quite low, we can observe that the baseline model, which is a deep neural network, cannot train at all with just a few training images.

### 4.2 CIFAR10 dataset

The model for classifying the CIFAR10 dataset is the 32-layer ResNet model [8], and 60,000 of CIFAR10 images are used for training, while 10,000 images are used for validation. The optimizer used for model training is Adam [39], the training epoch is 120, learning rate is 0.001, and batch size is 128. As a result of training, the accuracy of the CIFAR10 dataset classification model is 91.41 %. For FGSM and BIM, $\varepsilon$ is set to 9/255. Hyperparameters for the other attacks are the same as for the previous MNIST experiment.

**Table 3** shows the adversarial detection performances of the baseline model and our proposed model in experiments using CIFAR dataset. Experimental results show that the Gaussian process regression-based detector shows better detection performance than the baseline, except for the JSMA attack. Due to the small training dataset, the baseline model cannot train at all to detect the BIM, DeepFool, and C&W attacks. Also, for the C & W attack, which is considered the most powerful attack, the proposed method shows higher detection accuracy than the baseline model in both MNIST and CIFAR10 datasets.

### 4.3 Extremely small adversarial examples

Due to the nature of the Gaussian process regression, the proposed detection method can achieve high detection performance with only a small number of adversarial examples. To demonstrate this, we performed experiments to measure the performance of the detector by reducing the number of adversarial examples used in the detector training. The combinations of the dataset and adversarial attack used in the experiments are the MNIST dataset-FGSM attack and CIFAR10 dataset-JSMA, where the performance of the proposed detector is lower than the baseline model in the experiments using 300 adversarial examples in the training dataset. Without changing the model structure or other hyperparameters, we changed the number of adversarial examples used in model training to (300, 200, 100, 50, 30, and 10), and set natural images to the same number as the adversarial examples.

**Fig. 5** shows the performance of the proposed detection method and the baseline method according to the number of adversarial examples in the training dataset. When 300 adversarial examples were used for training, the baseline model performed better than the Gaussian process regression-based detector, but when training with fewer adversarial examples, the performance of our detector was higher than that of the baseline model. In particular, compared to the baseline model, our detector showed less variability in accuracy as the number of adversarial examples in the training dataset decreased. Thus, the Gaussian process regression-based detector can operate at high performance in environments with a small number of adversarial examples.

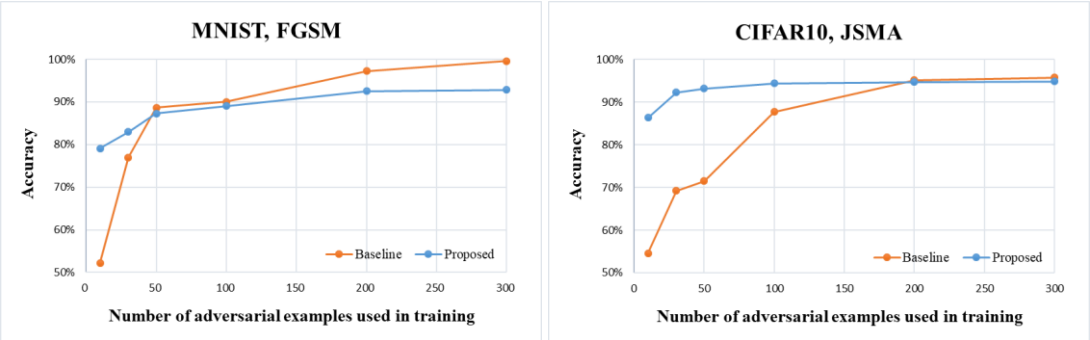


**Fig. 5**. Gaussian process regression-based adversarial image detector.

## 5. Conclusion

In this paper, we proposed the Gaussian process regression-based adversarial detection method. The proposed method first extracts the intermediate feature for a given input image from a pre-trained classification model, and then performs adversarial detection by the Gaussian process regression-based detector that has been trained with the extracted

low-dimensional information of images. Since Gaussian process regression expresses the correlation information between data by covariance matrix and performs regression based on this information, it can show high performance with only a small number of observed data.

The experimental result shows that our model demonstrates higher performance than the deep learning-based detection model for a small number of adversarial images. In particular, the proposed detector shows less accuracy variation on the number of adversarial examples in the training dataset than do deep learning-based detection models. Therefore, the Gaussian process regression-based detector can perform detection with high performance in the case of having a small number of adversarial examples, that is, when there is little information about the attack performed by the attacker. In future work, we plan to improve the performance of our detector by reflecting the characteristics of the adversarial image generated by the FGSM and BIM attacks.

## References

[1] Yoongyu Lim and Jee-Hyong Lee, "Balanced Cost-assigning Neural Networks for Imblanaced data," in *Proc. of 2018 Int. Conf. on Fuzzy Theory and Its Applications*, pp. 180-183, November 14-17, 2018.

[2] Hye-Woo Lee, Noo-ri Kim and Jee-Hyong Lee, "Deep Neural Network Self-training Based on Unsupervised Learning and Dropout," *Int. Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1, pp. 1-9, March, 2017. Article (CrossRef Link).

[3] Kyungtae Kim and Jee-Hyong Lee, "Predictive Models for Customer Churn using Deep Learning and Boosted Decision Trees," *Journal of Korean Institute of Intelligent Systems*, vol. 28, no. 1, pp. 7-12, February, 2018. Article (CrossRef Link).

[4] L. Zhang, J. Jia, Y. Li, W. Gao and M. Wang, "Deep Learning based Rapid Diagnosis System for Identifying Tomato Nutrition Disorders," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 4, pp. 2012-2027, April, 2019. Article (CrossRef Link).

[5] S. Naseer and Y. Saleem, "Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, pp. 5159-5178, October, 2018. Article (CrossRef Link).

[6] Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision," in *Proc. of 2010 IEEE Int. Symposium on Circuits and Systems*, pp. 253-256, May 30-June 2, 2010. Article (CrossRef Link).

[7] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. of Advances in Neural Information Processing Systems*, December 3-8, 2012. Article (CrossRef Link).

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770-778, June 27-30, 2016. Article (CrossRef Link).

[9] Y. Chen, F. Zhang and W. Zuo, "Deep Image Annotation and Classification by Fusing Multi-Modal Semantic Topics," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 392-412, January, 2018. Article (CrossRef Link).

[10] H. Sima, A. Mi, X. Han, S. Du, Z. Wang and J. Wang, "Hyperspectral Image Classification via Joint Sparse representation of Multi-layer Superpixles," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, pp. 5015-5038, October, 2018. Article (CrossRef Link).

[11] Min-Sub Won and Jee-Hyong Lee, "Embedding for Out of Vocabulary Words Considering Contextual and Morphosyntactic Information," in *Proc. of 2018 Int. Conf. on Fuzzy Theory and Its Applications*, pp. 212-215, November 14-17, 2018. Article (CrossRef Link).

[12] Hyunsoo Lee, Noo-ri Kim and Jee-Hyong Lee, "Attention Reader Model for Abstractive Text Summarization," in *Proc. of 13th Asia Pacific Int. Conf. on Information Science and Technology (APIC-IST 2018)*, pp. 13-15, June 24-27, 2018.

[13] YunSeok Choi, DaHae Kim and Jee-Hyong Lee, "Abstractive summarization by neural attention model with document content memory," in *Proc. of 2018 Conf. on Research in Adaptive and Convergent Systems*, pp. 11-16, October 9-12, 2018. Article (CrossRef Link).
[14] K. Al-Sabahi, Z. Zuping and Y. Kang, "Latent Semantic Analysis Approach for Document Summarization Based on Word Embeddings," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 1, pp. 254-276, January, 2019. Article (CrossRef Link).
[15] Noo-ri Kim, YunSeok Choi, HyunSoo Lee, Jae-Young Choi, Suntae Kim, Jeong-Ah Kim, Youngwha Cho and Jee-Hyong Lee, "Detection of document modification based on deep neural networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, issue 4, pp. 1089-1096, August, 2018. Article (CrossRef Link).
[16] Jina Kim and Jee-Hyong Lee, "Dual RNNs using Topic and Syntactic Information for Word Prediction," in *Proc. of 12th Asia Pacific Int. Conf. on Information Science and Technology (APIC-IST 2017)*, pp. 1-4, June 25-28, 2017.
[17] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Proc. of Advances in Neural Information Processing Systems*, December 7-12, 2015. Article (CrossRef Link).
[18] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 779-788, June 26-July 1, 2016. Article (CrossRef Link).
[19] I.J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proc. of Int. Conf. on Learning Representations*, May 7-9, 2015.
[20] A. Kurakin, I.J. Goodfellow and S. Bengio, "Adversarial examples in the physical world," in *Proc. of Int. Conf. on Learning Representations*, April 24–26, 2017.
[21] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *Proc. of 1st IEEE European Symposium on Security and Privacy*, pp. 372–387, March 21-24, 2016. Article (CrossRef Link).
[22] S.M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2574–2582, June 27-30, 2016. Article (CrossRef Link).
[23] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 39-57, May 22-26, 2017. Article (CrossRef Link).
[24] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks," *arXiv preprint arXiv:1511.04508*, November, 2015. Article (CrossRef Link).
[25] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu and J. Zhu, "Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1778-1787, June 19-21, 2018. Article (CrossRef Link).
[26] K. Grosse, P. Manoharan, N. Papernot, M. Backes and P. McDaniel, "On the (Statistical) Detection of Adversarial Examples," *arXiv preprint arXiv:1702.06280*, October, 2017.
[27] J. H. Metzen, T. Genewein, V. Fischer and B. Bischoff, "On Detecting Adversarial Perturbations," in *Proc. of Int. Conf. on Learning Representations*, April 24–26, 2017.
[28] N. Liu, H. Yang and X. Hu, "Adversarial Detection with Model Interpretation," in *Proc. of 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1803-1811, August 19-23, 2018. Article (CrossRef Link).
[29] Z. Gong, W. Wang and W.S. Ku, "Adversarial and Clean Data Are Not Twins," *arXiv preprint arXiv:1704.04960*, April, 2017.
[30] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," in *Proc. of 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3-14, November 3–3, 2017. Article (CrossRef Link).
[31] M. Ebden, "Gaussian Processes for Regression: A Quick Introduction," *arXiv preprint arXiv:1505.02965*, August, 2015.
[32] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *arXiv preprint arXiv:1712.07107*, July, 2018. Article (CrossRef Link).

[33] Byeongho Heo, Minsik Lee, Sangdoo Yun and Jin Young Choi, "Knowledge Distillation with Adversarial Samples Supporting Decision Boundary," *arXiv preprint arXiv:1805.05532*, May, 2018.
[34] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, et al., "Adversarial Attacks and Defences Competition," *arXiv preprint arXiv:1804.00097*, March, 2018. Article (CrossRef Link).
[35] C.E. Rasmussen, "Gaussian Processes in Machine Learning," *Advanced Lectures on Machine Learning. ML Summer Schools 2003. Lecture Notes in Computer Science*, vol. 3176, pp 63-71, Springer, Berlin, Heidelberg, 2003. Article (CrossRef Link).
[36] H. Nickisch and C.E. Rasmussen, "Approximations for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, vol. 9, pp. 2035-2078, October, 2008. Article (CrossRef Link).
[37] J. Snoek, H. Larochelle and R.P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Proc. of Advances in Neural Information Processing Systems*, December 3-8, 2012. Article (CrossRef Link).
[38] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv preprint arXiv:1212.5701*, December, 2012.
[39] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, December, 2014.

**Sangheon Lee** received the B.S. degree in Computer Engineering in 2018 from Sungkyunkwan University, Suwon, Korea. He is currently master course student in Department of Electrical and Computer Engineering at Sungkyunkwan University. His current research interests include deep-learning, adversarial defense, image classification and natural language processing.

**Noo-ri Kim** received the B.S. in Computer Engineering from Sungkyunkwan University, Suwon, Korea in 2013. He is currently pursuing his M.S.-Ph.D. in Computer Engineering at Sungkyunkwan University. His research interests include adversarial defense, recommender systems, text mining, and machine learning.

**Youngwha Cho** is currently a distinguished visiting professor in the college of software at the Sungkyunkwan University, Korea. He received his B.S. degree in statistics and the M.S. degree in computer science from Sungkyunkwan University in 1977 and 1990, respectively. In 1999, he completed his Ph.D. degree in computer science from Chungbuk University, Korea. He was with KISTI(Korea Institute of Science and Technology Information) as a President from 2001 to 2006. He has also served as a President at KISTEP(Korea Institute of S&T Evaluation and Planning) from 2007 to 2008. He joined Kyungwon University as a visiting professor from 2008 to 2010. His current research interests cover software engineering, data base, information communication technology, R&D strategies and so on.

**Jae-Young Choi** is a professor with the department of computer engineering, college of software at the Sungkyunkwan University, Korea. He received his B.S. degree in mathematics in 1995, and the M.S. and Ph.D. degrees in computer science from the Kyungwon University, Korea, in 1999 and 2004, respectively. From 2004 to the middle of 2006, he joined the Vision Laboratory at the University of California, Los Angeles, USA, as a postdoctoral researcher. He has also served as a BK21 research professor at Kyungwon University from 2006 to 2010. His research interests include computer vision, machine learning, ubiquitous computing, network management, software engineering and R&D strategies.

**Suntae Kim** is an Associate Professor of the Department of Software Engineering at Chonbuk National University. He received his B.S. degree in computer science and engineering from Chung-Ang University in 2003, and the M.S. Degree and PH.D. Degree in computer science and engineering from Sogang University in 2007 and 2010. He worked in Software Craft Co. Ltd., as a senior consultant and engineer for financial enterprise systems during 2002?2004. Also, he developed Android based Smart TV middleware from 2009 to 2010. His research focuses on software architecture, design patterns, requirements engineering, and source code mining.

**Jeong-Ah Kim** received Ph. D degree at ChungAng University. Since 1996, she has worked at Catholic Kwandong University as professor. She is the member of Korea Institute of Information Science and Engineering and the member of board of directors of Convergent Research Society. Her research areas are software product line engineering, software modeling, software process improvement, clinical decision support system.

**Jee-Hyong Lee** received his B.S., M.S., and Ph.D. in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1993, 1995, and 1999, respectively. From 2000 to 2002, he was an international fellow at SRI International, USA. He joined Sungkyunkwan University, Suwon, Korea, as a faculty member in 2002. His research interests include fuzzy theory and application, intelligent systems, and machine learning.