Security of Deep Learning Systems: Defense Against Adversarial Attacks Based on Cryptographic Principles

CAP 5150 Fall 2022 - Final Report

Alina Ageichik
UCF CECS Graduate Student
4000 Central Florida Blvd
Orlando, FL 32816
+1 (407) 823 2455
aageichik@knights.ucf.edu

Christian Kansley
UCF CECS Graduate Student
4000 Central Florida Blvd
Orlando, FL 32816
+1 (407) 823 2455
ckansley@knights.ucf.edu

Sebastian Quiroga
UCF CECS Graduate Student
4000 Central Florida Blvd
Orlando, FL 32816
+1 (407) 823 2455
squiroga@knights.ucf.edu

Shiyi Gong
UCF CECS Graduate Student
4000 Central Florida Blvd
Orlando, FL 32816
+1 (407) 823 2455
shiyigong@knights.ucf.edu

ABSTRACT

This report is intended to present the techniques, challenges, and progress this team has made during the project throughout the semester. We reproduce the method of defense against adversarial attacks on the deep learning image classification systems called a key-based diversified aggregation (KDA). This defense method was inspired by the second Kerckhoff's cryptographic principle that assumes that the attacker (i) knows the architecture of the classifier and the used defense strategy, (ii) has access to the training data, and (iii) does not know a secret key used for defense and does not have the access to the trainable parameters of the system. The KDA randomization with key-based sign flipping is designed to achieve the robustness of the system and is addressed in this research to more accurately and precisely come to the intended conclusion. The randomization is performed on multiple channels simultaneously and the secret is shared between training and testing stages, which gives an advantage to the defender. We test the proposed defense strategy on three state-of-the-art adversarial attacks in both grey-box and black-box scenarios. We fully train vanilla classifiers and multi-channel with KDA classifiers, attack those models, and test the ability of those models to defend against these attacks.

Keywords

Cryptography; deep learning; machine learning; adversarial attack; computer vision; malware; secret keys; classification.

1. Introduction

The deep learning field has been a groundbreaking subset of machine learning, and machine learning is well-known to be a subset of artificial intelligence. It is an ever-evolving field that umbrellas over the artificial intelligence concept. They both go hand in hand, machine learning allows for artificial intelligence or any type of software to learn without supervision. Through the input of different historical data, the machine becomes smarter at predicting outcomes making the software you're using applicable to different industries.

Machine learning has its roots as a mathematical model for neural networks. It comes from a paper written by logician Walter Pitts and neuroscientist Warren Mchulloch in 1943. In this paper, they were trying to map human cognition through mathematics. After the discovery of machine learning and AI through neural mapping, Alan Turing developed a test that if an AI can prove to a person that it is human it passes. After these two significant events, AI and machine learning began being tested in various ways to reach and know the different capabilities machine learning and AI had. Various neural networks make up the backbone of deep learning algorithms and systems. Neural networks achieve state-of-the-art performance in many areas such as computer vision, natural language processing, speech recognition, and robotics..

Regarding the security aspect, a lot of money is going into the deployment of machine learning-based systems due to their ease of use. However, many deep learning algorithms and systems have security vulnerabilities that can be used to perform adversarial attacks. Adversarial attacks aim to add a perturbation to the original input for deep learning systems that can trick it and lead to making an incorrect decision. Adversarial attacks question the security of deep learning systems, as well as their adequate usability. In this paper, we concentrate on deep learning systems in the image domain such as systems based on neural network image classification. We will be focusing on how security, cryptography, and machine learning go hand in hand. We aim to reproduce and improve the paper [1] that uses the second Kerckhoff's cryptographic principle as inspiration. The paper explores the method called key-based diversified aggregation (KDA) mechanism as a defense strategy in both gray-box and black-box scenarios. The gray-box and black-box scenarios are more realistic types of attack scenarios rather than a white-box. The gray-box attacks assume that the attacker has some knowledge about the model but there is some unknown element, or limited access to the intermediate results. The black-box attack assumes that the attacker only sees the output of the model without having any knowledge about the system's architecture and internal parameters. The KDA assumes that the attacker (i) knows the architecture of the classifier and the used defense strategy, (ii) has access to the training data, and (iii) does not know a secret key used for defense and does not have the access to the trainable parameters of the system. By using the proposed method, the defender gains the following advantages over the attacker (i) information advantage due to the usage of a secret key (ii) because it is a multi-channel system, the attacker must attack at least several channels simultaneously to succeed (iii) limited access to the proposed architecture does not allow the attacker to build a bypass system. (iv) the right choice of the aggregation operator and channels at random increases the security of the

system (v) randomness of each channel can be adjusted, which gives a possibility for adaptation for different attacks. Figure 1 demonstrates the overall idea.

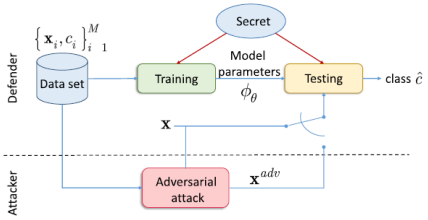

Figure 1. The information access diagram: the defender has an access to the training data and secret shared between the training and test stages while the attacker has only access to the shared training data set

2. Related Work

This project is based mainly on the following papers: Taran, Olga., et al. (2020) [1], Taran, Olga., et al. (2019) [2], Taran, Olga., et al. (2019) [3], and Taran, Olga., et al., (2018) [4]. The earliest paper [4] explores how deep learning architectures are vulnerable to adversarial examples. It tackled this problem by introducing the second Kerckhoff's cryptographic principle and how it can be integrated to build a defense system against adversarial attacks on the deep learning image classification systems. They introduced the concepts of a secret key integrated into a deep learning image classification system, requirements to secret elements, and data-independence of security imposing transformation. The attacks used to test their defense method are the Fast Gradient Sign Method (FGSM) [8] and the Carlini&Wagner (C&W) [5]. The next paper [3] explores the key based diversified aggregation (KDA) multi-channel system and how it can be used as a defense against different adversarial attacks in both black-box and grey-box scenarios. They introduce aggregation operator and pre-filtering. They tested their defense method on two state-of-the-art adversarial attacks such as C&W and One Pixel [6]. The next paper [2] explores KDA multi-channel system, and integrated randomized diversification in grey-box scenario only. They tested their defense method on C&W adversarial attack. Those three papers are the predecessors for their latest paper [1]. The latest paper, the authors based their method on all aforementioned papers, and integrate all of the techniques used in their previous papers to create a single system. In addition, they use both grey-box and black-box scenarios, and test their entire defense system on C&W, One Pixel and Projected Gradient Descent [7] attacks.

3. Attack and Defense Methods

In general case, for an input image $x \in R^{N\times S}$ with a class label $c \in \{1, 2, \dots, M_c\}$, the optimization problem of finding an adversarial example with the additive perturbation $x_{adv} = x + \varepsilon$ and a target class c_{adv} can be formulated as

$$\begin{aligned} &\min_{\epsilon} \ \mathcal{L}\left(c^{adv}, \phi_\theta(\mathbf{x}+\boldsymbol{\epsilon})\right) + \lambda\|\boldsymbol{\epsilon}\|_p, \\ &s.t. \quad \mathbf{x}+\boldsymbol{\epsilon} \in [0,1]^{N\times S}, \end{aligned} \tag{1}$$

where L(.) is a classification loss, φ_0 is a targeted classifier, $c \neq c_{adv}$, λ is a Lagrangian multiplier, and l_p-norm is defined as:

$$\|\boldsymbol{\epsilon}\|_p = \left(\sum_{i=1}^{N\times S} |\epsilon_i|^p\right)^{\frac{1}{p}},$$
$$\text{with } 0 \le p \le 2.$$

3.1 C&W Attack

The gradient-based C&W attack proposed by Carlini and Wagner in 2017 has the following mathematical formulation:

$$\begin{aligned} &\min_{\epsilon} \ a \cdot f(\mathbf{x}+\boldsymbol{\epsilon}) + \|\boldsymbol{\epsilon}\|_p, \\ &s.t. \quad \mathbf{x}+\boldsymbol{\epsilon} \in [0,1]^{N\times S}, \end{aligned} \tag{2}$$

where a > 0 is a suitably chosen constant, f (.) is the new objective function such that $\varphi_0 (x + \varepsilon) = c^{adv}$, if and only if $f(x + \varepsilon) \le 0$. The authors investigated a few objective functions f (.), and as the most efficient one:

$$f(\mathbf{x}^{adv}) = \max\left(\max_{l\neq c^{adv}}\left(Z\left(\mathbf{x}^{adv}\right)_l\right) - Z\left(\mathbf{x}^{adv}\right)_{c^{adv}}, -\kappa\right), \tag{3}$$

where l is an index of any class, c^{adv} is an index of the adversarial class, $Z(x) = \varphi_{\theta n-1}(x)$ is the result of the network φ_0 before the last activation function, and κ is a constant that controls the confidence of the attack.

3.2 PGD Attack

The non-gradient based PGD attack proposed by Madry et at. in 2017 is an iterative version of FGSM attack that solves the optimization problem (1) by computing an adversarial example at the iteration t + 1 as:

$$\mathbf{x}_{t+1}^{adv} = Proj\left(\mathbf{x}_t^{adv} + \alpha \cdot sign\left(\nabla_{\mathbf{x}}\mathcal{L}\left(c^{adv}, \phi_\theta(\mathbf{x}_t^{adv})\right)\right)\right) \tag{4}$$

where Proj(.) keeps x^{adv} t+1 within a predefined perturbation range and valid image range, α a is the magnitude of the adversarial perturbation in each iteration.

3.3 One Pixel Attack

Differential Evolution (DE) optimization algorithm [9] for the attack generation. The DE algorithm does not assume that the objective function is known or differentiable, it observes the output of the classifier as a black-box. The attack aims at perturbing a predefined number of pixels in the input image $x \in R^{N\times S}$. This optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\epsilon} \ \mathcal{L}\left(c^{adv}, \phi_\theta(\mathbf{x}+\boldsymbol{\epsilon})\right), \\ &s.t. \quad \|\boldsymbol{\epsilon}\|_0 \le d, \end{aligned} \tag{5}$$

where d is the number of pixels to be perturbed and L(.) is a classification loss.

3.4 Classification based on KDA

The diagram of the method is shown in Figure 2. The KDA has six main blocks:

1. *Pre-filtering* $\phi_\beta(x)$ that has an optional character. This block returns the input image x by removing high-magnitude outliers made by the attacker. The variety of pre-filtering algorithms can be used such as a simple local mean filter or more complex ones such as BM3D [10] or based on DNN mappers [11].
2. *Pre-processing* of the input data by mapping the transform W_j, $1 \le j \le J$. The transform W_j can be any linear data-independent mapper such as the random projection with the dimensionality reduction or expansion or Discrete Fourier Transform, Discrete Cosines Transform etc. The transform W_j can also be a learnable transform, but the data independent W_j is preferred to avoid the leakage about it from the training data. As a final note, W_j can be based on a secret key k_j.

3. *Data-independent processing* P_{ji}, $1 \le i \le I$ represents the randomization part that serves as a defense against gradient back propagation. Figure 3 shows several cases. In Figure 3a, $P_{ji} \in \{0, 1\}^{l \times n}$, where $l < n$, is a lossy sampling of the input image of length n. In Figure 3b, $P_{ji} \in \{0, 1\}^{n \times n}$, is a lossless permutation. In Figure 3c, $P_{ji} \in \{-1, 0, +1\}^{n \times n}$, is a sub-block sign flipping. The key defined region of key-based sign flipping is highlighted. This operation is reversible, therefore, lossless for an authorized party. As a final note, if we want to make the data-independent processing irreversible for the attacker, we must use a P_{ji} based on secret key k_{ji}.

4. *Classification* block $\varphi_{\theta ij}$ can be any classifier or ensemble of classifiers. However, if the classifier is designed for the classification in the direct domain, then it is preferable that it is preceded by W_j^{-1}. The main concern here is whether to use the convolutional layer or fully connected ones.

5. *Classifiers' selection* S with a key k_s assumes to randomly select J_s outputs of classifiers out of JI outputs of pre-trained classifiers for a further aggregation.

6. *Aggregation* block A_ϑ can be represented by any operation, for example, a simple summation to learnable operators adapted to the data or a particular adversarial attack.

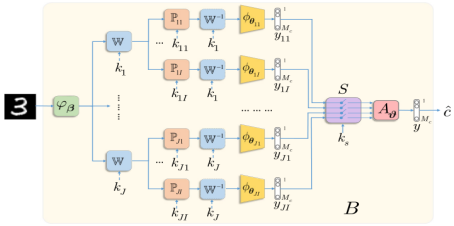

Figure 2. Generalized diagram of the proposed multi-channel system with the KDA.

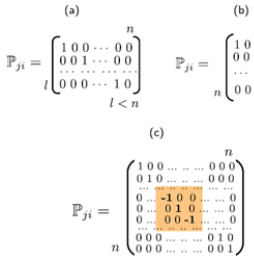

Figure 3. Randomized transformation P_{ji}, $1 \le j \le J$, $1 \le i \le I$, a) randomized sampling, b) randomized permutation, c) randomized sign flipping in the sub-block defined in orange. All transforms are key-based.

At Figure 2, we see that blocks 2,3, and 4 represent the chain of processing in a parallel multi-channel structure followed by the classifiers and the aggregation blocks. The class determined based on the aggregated result. There is a possibility for the attacker to use the full system as white box by accessing the intermediate results inside. The attacker can discover the secret keys k_j and/or k_{ji} and make the system differentiable by using the Backward Pass Differentiable Approximation technique [12] or by replacing the key-based blocks using the "bypass" mappers. Therefore, we want to highlight the importance of restricting the access to the intermediate results within the block B that satisfies the black-box scenario, and the Kerckhoffs's cryptographic principle where by assumption, the algorithm and system's architecture are known, besides the used secret key that corresponds to the various secret perturbations. The training process can be mathematically defined as follows:

$$(\hat{\vartheta}, \{\hat{\theta}_{ji}\}) = \arg\min_{\vartheta, \{\theta_{ji}\}} \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{i=1}^{I} \mathcal{L}(c_t, A_\vartheta(\phi_{\theta_{ji}}(\mathbb{W}_j^{-1} \mathbb{P}_{ji} \mathbb{W}_j \varphi_\beta(\mathbf{x}_t)))), \tag{6}$$

In the proposed system, there are a few simplifications made to gain the advantage of the defender over the attacker.

1. Training is performed per channel independently up to selection and aggregation. At testing phase, pre-trained classifiers are chosen for the aggregation by the defender, so the attacker must target a subset of classifiers to influence the final decision of the classifier(s). It won't be possible for the attacker to use a single perturbation to trick all classifiers simultaneously.
2. The goal of the data-independent processing P_{ji} is to prevent a gradient back propagation into the direct domain, but the training is adapted to P_{ji} in each channel.
3. As an aggregation operator, it can be an additional classifier that takes the soft outputs of multi-channel classifiers as input, and outputs the final prediction. The majority voting or summation of the multi-channel outputs with the maximum class selection.
4. Due to independent randomization in each channel, the security of the system increases. Each channel can have the adjustable level of randomization, which gives the advantage. Regarding a one-channel system, the level of randomness can be either insufficient or too high, that drops classification accuracy.

3.5 Randomization with Key-Based Sign Flipping in the DCT Domain

In the proposed system shown in Figure 2, one of the main elements is the randomized diversification of the input image using data-independent processing P. The permutation of pixels is one case of such a diversification. However, the performance of such a defense degrades because of the high sensitivity to the gradient perturbations. Let W be the DCT operator and the local sign flipping $P_{ji} \in \{-1, 0, 1\}^{n \times n}$ based on the individual secret key k_{ji} for each classifier φ_θ. Local refers to the processing only in some sub-band (block) of the image. The length of k_{ji} is the length of the corresponding sub-band i.e., n×n. An image can be split into overlapping or non-overlapping sub-bands of different sizes and positions that we keep secret. An image in the DCT domain is split into four nonoverlapping fixed sub-bands of the same size: (L) top left i.e., low frequencies of the image, (V) vertical, (H) horizontal, and (D) diagonal sub-bands as shown in Figure 4. In V, H, and D sub-bands, the key-based sign flipping is applied independently while keeping other sub-bands unmodified. The length of a secret key within each sub-band is n × n = image size/2 × image size/2. The effects of such sign flipping are barely noticeable.

Figure 4. Local key-based sign flipping in the DCT sub-bands: a) sub-bands, b) original image, c) with a sign flipping in V, d) with a sign flipping in H, and e) image with a sign flipping in D.

Figure 4 shows the corresponding multi-channel architecture. As an aggregation operator A_9, a simple summation was used, and the selector S uses the outputs of all classifiers JI. As a pre-filtering ϕ_β, a filter based on a difference of the point of interest in the center of the window with the median value in the window of size 3×3 around this point was used. If the magnitude of difference exceeds a predefined threshold, we consider such a pixel to be corrupted by the adversary, so a mean value computed in the window replaces the value of such a pixel, otherwise, it is kept intact. Under such perturbation, we can mathematically define the training of each classifier $\varphi_{\theta ji}$ independently as follows:

$$\hat{\theta}_{ji} = \arg\min_{\theta_{ji}} \sum_{t=1}^{T} \mathcal{L}(c_t, \phi_{\theta_{ji}}(\mathbb{W}^{-1}\mathbb{P}_{ji}\mathbb{W}\,\varphi_\beta(\mathbf{x}_t))). \qquad (7)$$

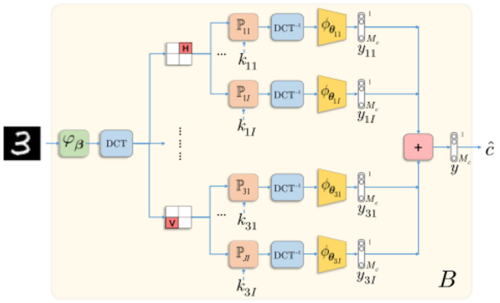

Figure 4. Multi-channel classification with the local DCT sign flipping.

4. Results

4.1 Limitations

Training, testing, and deployment of deep learning image classification models may take a few days and up to a few weeks due to their high mathematical complexity and large amount of image data. This is applicable to modern adversarial attack methods when it comes to the process of the generation of adversarial images for the purpose of testing the defense mechanisms of deep learning models. Hardware appears to be our current limiting factor as not all of the group members have immediate access to GPU clusters. It was earlier addressed in our project proposal and a previous milestone that we have only two group members with a GPU available. Using a single GPU might not be enough to conduct timely training and testing. Performing computations on CPU doubles the time, so it is not a good option, unless it is a back up plan. We found a source code for the paper on GitHub (https://github.com/taranO/multi-channel-KDA), and we were able to reproduce the entire paper, run code to get the results for all attacks, and test the proposed defense mechanism. However, we modified and debugged the source code as the Python version and all libraries were outdated i.e. implemented in Python 2 with corresponding versions of all libraries. We used Python 3.8.5 and latest versions of deep learning libraries such as PyTorch, Keras and TensorFlow. We were able to fully train and test two baseline or "vanilla" classifiers and a few multi-channel KDA classifiers with various parameters such as the number of channels and permutations, and different sub-bands. Then, were able to perform three state-of-the-art attacks stated in the paper such as the One Pixel, C&W and PGD attacks on image datasets and generate adversarial images, then feed those perturbed images to all classifiers as input, test the proposed defense mechanisms and transferability. To train vanilla classifiers and multi-channel classifiers, we spent about 1.5 weeks. To perform a One Pixel attack, we spent additional 2 weeks. To perform C&W, we spent about 3 weeks, and to perform PGD, we spent about 5 days.

4.2 Attack Scenarios

As we stated before, the main concept that the proposed system is based on consists in an information advantage over the attacker, the attacker. That is, the attacker has a limited access to the intermediate results and does not know secret keys within the system. Therefore, we test the efficiency of the multi-channel architecture with the diversification and randomization by the key-based sign flipping in the DCT domain against the adversarial attacks in the following three scenarios:

1. Gray-box transferability attacks from a single-channel model to a multi-channel model tested on (i) the C&W with the constraints on l2, l0, and l∞ norms and (ii) the PGD attack.
2. Gray-box transferability attacks from a multi-channel model to a multi-channel model under different keys tested on the OnePixel attack with perturbation in 1, 3, and 5 pixels.
3. Black-box direct attacks tested on the OnePixel attack with perturbation in 1, 3, and 5 pixels.

4.3 Training Set Up and Results

4.3.1 Metrics

To measure a classification ability of all classifiers, we use the following formula to calculate the accuracy of the predictions.

$$Accuracy = \frac{correct}{total} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP stands for True Positives. That is, the number of samples that the classifier correctly predicted the positive class. TN stands for True Negatives. That is, the number of samples that the classifier correctly predicted the negative class. FP stands for False Positives. That is, the number of samples that the classifier incorrectly predicted the positive class. FN stands for False Negatives. That is, the number of samples that the classifier icorrectly predicted the negative class.

To measure the defense abilities of our models, we used a classification error, which is the following:

$$Classification\ Error = \frac{errors}{total} = \frac{FP+FN}{TP + TN + FP + FN}$$

The authors used classification error metrics to measure all results.

4.3.2 One Pixel Attack

We trained the VGG16 [13] and ResNet18 [14] baseline models during 100 epochs with learning rate 1e-3, weight decay 5e-4, and a batch size of 128 using NVIDIA GeForce RTX 2080 Super Max-Q 8GB. For the VGG16 model, the Stochastic Gradient Descent optimizer was used, and the Adam optimizer was used for the ResNet18 model during the training along with the Cross Entropy loss function for both. We also trained a multi-channel system with KDA using the aforementioned base models and all the corresponding parameters, and loss functions for each classifier.

4.3.3 C&W Attack

We trained the same models VGG16 and ResNet18 with different set of parameters such as the learning rate of 1e−2 and weight decay of 1e−6, and a batch size of 128. The SGD optimizer was used to train both the "attacked vanilla" and "transferability

vanilla" models for 50 epochs only because after 50th epoch, the saturation was observed. In the multi-channel model, each classifier was trained using the learning rate of 1e−3, weight decay of 1e−6, and a batch size 64 during 100
epochs using Adam optimizer. For the l2-constraint attack, we used the same classifiers with the learning rate of 1e-2, confidence level of 0, maximum number, of iterations 1000 with early stopping if the gradient descent freezes, and the minimum and maximum pixel values of −0.5 and 0.5 correspondingly. For the l0 and l∞ attacks, the constant factor was 2, and the rest of the used parameter was the same as in case of 2 attack.
All models were trained using a CIFAR-10 dataset [6] of 50000 color images of size 32 × 32 with 10 classes. See Figure 3 for the examples. The results of the training of all models are presented in Table 1 and 2. We can observe that the accuracy indeed drops as we apply a multi-channel KDA defense mechanism on top of base classifiers However, as the authors stated, the paper aims in providing a robust defense mechanism against adversarial attacks rather than achieve the highest classification accuracy. Also, the authors stated that the classification accuracy increased with the number of channels, but from our results, we can't confirm that for all models.

Figure 3. CIFAR-10 dataset image samples

4.3.4 PGD Attack

The PGD attack was used to attack the VGG16 and ResNet18 models with parameters of α equals to 0.5, step size of 0.01, and 100 iterations.

4.4 Testing Set Up

Although a CIFAR-10 testing dataset has 10000 images, we only used the first 1000 images for testing just like the authors as the testing process also takes a sufficient amount of time. We generated 1000 adversarial samples using OnePixel attack.

5. Results and Discussion

First of all, we would like to mention that all the values we obtained deviate a little from the values that the authors of the proposed method obtained. We believe that it happened due to the difference in the version of Python, and all corresponding libraries. That is, most likely a round-off error. The authors made certain conclusions that are not universal from our experiments, but still, confirm the effectiveness of their proposed method.

5.1 Trained Models

Tables 1 and 2 show the results of the classification of VGG16 and ResNet-18 base vanilla and KDA models. The authors did not provide these results, but we do provide them for more understanding of the classification process and results. As the authors mentioned, the purpose of the method is to defend the deep learning architectures and not increase the classification accuracy, but to reduce the classification error. Therefore, we can confirm that the classification accuracy of vanilla classifiers is higher than the one of KDA, but only a couple of percent. Also, the increase in the number of channels does not really affect the classification accuracy.

5.2 Gray-box Transferability

5.2.1 Multi-channel to a Multi-channel

In this scenario, we assume that the attacker knows the architecture of a multi-channel classifier with the proposed defense strategy, as well as, has access to the same training data as the defender. The only thing that the attacker does not know is the defender's secret keys used to build the defense mechanism. The attacker trains his multi-channel classifier using some specific set of keys and produces the adversarial examples while the defender trains the similar system but using different keys and different model's parameters as a part of the secret. The results for the gray-box transferability of the adversarial examples from one multi-channel to another multi-channel model using different keys can be observed in Table 3 for the OnePixel attack with perturbation in 1, 3, and 5 pixels respectively. From the results we can confirm that the success of attack does not exceed 0.5% compared to the classification accuracy on the original non-attacked data.

5.2.2 Single-channel to Multi-channel

For C&W attack, the results are given in Table 6 with the constraints on l2, l0 and l∞ norms. For PGD attack, the results are given in the Table 7. The vanilla model for a single-channel was chosen to be known to the attacker, and both the attacker and defender have the access to training data. Let's say, the attacker trains the single-channel vanilla classifier and generates the adversarial examples against the system. In columns "attacked vanilla" in Tables 6 and 7 we can see the result of such attack. In Table 6, the classification error is very high for such a model meaning that the system ihas been hacked and the classifier can't make correct classifications almost 100%. At the same time, in Table 7 we see that the PGD attack is less efficient.

Let's consider a scenario where the defender trains the same single-channel model with the same training data set but with different parameters of the model. We can see those results in the same tables for C&W and PGD. In the "Transferability Vanilla" column, we can see the results of the transferability of adversarial images to the defender's single-channel classifier. We observe a very low classification error meaning that the proposed attacks are not efficient, and the defense works well. Finally, the "Transferability KDA" column shows us the results of the transferability of the same adversarial images to the multi-channel model with KDA. Here, the authors stated that the increase in the number of channels produces lower classification error, however, from our experiments, we can only partially confirm that. This is not universal, and we believe that it is due to a round-off error again. But, we observe that there is only about 2% of attack success in case of C&W with l0 constraint on CIFAR-10 dataset. We can observe the similar behavior for the PGD in case of the C&W l2 and C&W l∞ on MNIST and Fashion-MNIST datasets. There is about 1–3% of successful attacks while for the C&W l0 it's a bit higher, about 2.5–5.5%. The authors stated that it might be related to a high sparsity of the original images. Here, we can conclude that the multi-channel model is robust to the adversarial examples generated for the single-channel model with the same model's architecture.

5.3 Black-Box Direct Attack

Table 4 shows us the results obtained for the direct attacks to a single-channel and a multi-channel models in the black-box scenario using One Pixel attack. The row "Original" means that models used non-attacked, original data as input. In this scenario, the attacker does not know about the classifiers' architecture, number of channels, and used defense mechanisms. The attacker

can only see the predicted class label for the given input. We can see that One Pixel is efficient for the vanilla models. We confirm that the classification error is about 60-80%, and in the case of the ResNet18 model, it is about 35-60%.

5.4 Key-Based Aggregation

In addition to the multi-channel system with the fixed channels, the authors experimented with the similar system for the case when the channels were chosen based on a random key. They averaged results over 10 runs as given in Table 5, 7 and 8. Comparing the results for the KDA in Tables 5 and 8, we can indeed notice a small degradation of performance when selecting the random selection of channels. The authors point out that it is because the subbands chosen for the randomization in runs provided in Table 5 always correspond to the three main V, H, and D subbands, whereas the subbands representing channels in the setup of runs provided in Table 8were chosen at random.

Method		**Accuracy**		
	Channel	**3**	**6**	**9**
Original	73.29	-	-	-
Multi-channel (d1)	-	***70.71***	70.23	***70.84***
Multi-channel (d2)	-	70.10	70.51	70.14
Multi-channel (d3)	-	70.35	***71.77***	70.80
Multi-channel (h1)	-	67.92	68.14	68.79
Multi-channel (h2)	-	68.88	68.62	68.65
Multi-channel (h3)	-	67.84	69.04	68.32
Multi-channel (v1)	-	70.37	70.54	69.98
Multi-channel (v2)	-	69.12	99.72	69.29
Multi-channel (v3)	-	69.52	68.65	69.53

Table 1. Validation accuracies during training of VGG16. Letters d, h and v are subbands, numbers 1,2,3 next to subbands correspond to key-based flipping lossless permutation

Method		**Accuracy**		
	Channel	**3**	**6**	**9**
Original	84.75	-	-	-
Multi-channel (d1)	-	***82.78***	82.69	82.17
Multi-channel (d2)	-	82.63	***82.92***	***83.39***
Multi-channel (d3)	-	82.25	82.50	82.52
Multi-channel (h1)	-	81.24	80.92	80.80
Multi-channel (h2)	-	80.77	80.49	80.18
Multi-channel (h3)	-	80.73	80.74	80.43
Multi-channel (v1)	-	81.39	81.07	80.80
Multi-channel (v2)	-	81.06	81.01	81.43
Multi-channel (v3)	-	81.22	81.23	81.36

Table 2. Validation accuracies during training of ResNet-18. Letters d, h and v are subbands, numbers 1,2,3 next to subbands correspond to key-based flipping lossless permutation.

Data type	**KDA with different keys #channels #classifiers**		
	3	**6**	**9**
VGG16	-	-	-
Original	12.11	10.84	10.21
OnePixel p=1	***13.09***	10.56	***10.48***
OnePixel p=3	12.44	11.01	10.01
OnePixel p=5	12.56	***11.23***	10.32
ResNet18	-	-	-
Original	9.99	8.45	7.29
OnePixel p=1	9.12	8.89	7.13
OnePixel p=3	10.05	8.97	7.45
OnePixel p=5	10.59	8.33	8.02

Table 3. Classification error (%) on the first 1000 test sample (CIFAR-10) for the gray-box OnePixel transferability attacks from multi-channel model under different key

Data type	**Attacked vanilla**	**Attacked KDA #channels #classifiers**		
		3	**6**	**9**
VGG16		-	-	-

Original	10.02	10.78	9.01	8.89
OnePixel p=1	58.73	10.56	***9.44***	8.45
OnePixel p=3	71.75	10.94	8.98	8.34
OnePixel p=5	***79.78***	***12.01***	9.39	***9.40***
ResNet18		-	-	-
Original	9.01	***11.89***	9.01	7.52
OnePixel p=1	36.99	11.65	9.02	7.76
OnePixel p=3	49.45	11.21	9.05	***7.89***
OnePixel p=5	***59.89***	11.77	***9.13***	7.58

Table 4. Classification error (%) on the first 1000 CIFAR-10 test sample (CIFAR-10) for the direct black-box OnePixel attacks

Data type	Attacked KDA #channels #classifiers		
	3	**5**	**7**
VGG16	-	-	-
Original	11.27	9.82	9.21
OnePixel p=1	11.26	9.32	**9.49**
OnePixel p=3	11.58	9.66	9.02
OnePixel p=5	**11.89**	**10.43**	9.42
ResNet18	-	-	-
Original	11.21	9.78	8.72
OnePixel p=1	11.12	9.21	8.91
OnePixel p=3	***11.34***	9.77	8.79
OnePixel p=5	11.01	***9.81***	***9.19***

Table 5. Classification error (%) on the first 1000 test samples (CIFAR-10) for the multi-channel system against the direct black-box OnePixel attacks with randomly selected channels (the average results over 10 runs).

Data type	Attacked vanilla	Transferability vanilla	Transferability KDA #channels #classifiers		
			3	**6**	**9**
MNIST			-	-	-
Original	1	0.92	0.51	0.54	0.57
C&W ℓ2	99.23	6.13	4.56	4.89	4.76
C&W ℓ0	***99.84***	***14.5***	***7.23***	***7.6***	***6.34***
C&W ℓ∞	99.11	4.85	2.67	2.34	2.01
Fashion-MNIST					
Original	7.32	7.52	8.23	7.45	7.7
C&W ℓ2	***100***	11.06	9.34	8.51	8.91
C&W ℓ0	99.99	11.80	***10.38***	***9.32***	***10.04***
C&W ℓ∞	99.67	***11.98***	9.34	8.93	8.98
CIFAR-10					
Original	20.96	20.62	21.40	19.30	19.40
C&W ℓ2	99.56	25.03	22.59	21.01	***23.86***
C&W ℓ0	98.9	***31.01***	***24.89***	***23.04***	23.30
C&W ℓ∞	***100***	25.32	22.53	21.49	21.43

Table 6. Classification error (%) on the first 1000 test samples for the gray-box C&W transferability attacks from a single-channel model to a multi-channel model.

Data type	Attacked vanilla	Transferability vanilla	Transferability KDA #channels #classifiers			
			3	**5**	**7**	**9**
VGG16			-	-	-	-
Original	10.39	11.67	11.52	9.95	9.32	9.52
PGD	***16.03***	***15.47***	***14.26***	***12.02***	***11.81***	***11.38***
ResNet18			-	-	-	-
Original	9.47	10.34	11.21	9.78	8.85	8.73
PDG	***18.03***	***14.82***	***14.21***	***11.92***	***10.72***	***9.51***

Table 7. Classification error (%) on the first 1000 test samples (CIFAR-10) for the gray-box PGD transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs)

Data type	Attacked KDA #channels #classifiers		
	3	**5**	**7**
MNIST	-	-	-
Original	0.61	0.58	0.62
C&W ℓ2	4.99	4.82	4.01
C&W ℓ0	***7.83***	***7.21***	***7.72***
C&W ℓ∞	3.39	3.18	2.34
Fashion-MNIST	-	-	-
Original	8.47	8.22	8.4
C&W ℓ2	9.06	9.19	8.75
C&W ℓ0	***10.21***	***10.3***	***10.2***
C&W ℓ∞	9.39	9.18	9.08
CIFAR-10			
Original	21.3	20.59	20.11
C&W ℓ2	22.59	21.35	21.18
C&W ℓ0	***27.81***	***25.59***	***24.21***

C&W ℓ_∞	24.54	22.47	21.76

Table 8. Classification error (%) on the first 1000 test samples for the gray-box C&W transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs)

6. Work Distribution of Team Members

Alina Ageichik primarily ran the majority of experiments and developed the initial models, as well as, debugging Python libraries to work with the more up-to-date Python 3. Alina has trained all vanilla classifiers, and some KDA classifiers. She also performed One Pixel and PGD attacks on all of those models.Christian Kansley primarily worked on report formation and editing, as well as, assisting with other sections where requested. Sebastian Quiroga researched the literature needed for the base level understanding necessary for the project. Information such as policies, laws, ethics, historic background, and implications of defending against adversarial attacks. Shiyi Gong worked on the debugging of code as well, and ran other experiments with KDA and randomization key-based sign flipping, as well as, performed C&W attacks on those models.

7. Conclusion

In project, we addressed the security problems of deep learning image classifiers to defend against adversarial attacks in both grey-box and black-box settings. We reproduced the method for the defense against adversarial attacks on the deep learning image classification systems called a key-based diversified aggregation (KDA). This defense method was inspired by the second Kerckhoff's cryptographic principle. We were able to fully train vanilla classifiers and multi-channel with KDA classifiers, attack those models using three state-of-the-art attacks such as the One Pixel, C&W, and PGD, and test the ability of those models to defend against these attacks. We confirm that the proposed defense mechanisms provided a successful defense against all three adversarial attacks in both grey-box and black-box scenarios. The system is robust against (i) gray-box transferability attacks from a single-channel model to a multi-channel model under assumption that the attacker knows only the single-channel model architecture, (ii) gray-box transferability attacks from a multi-channel model to a multi-channel model trained under different keys under assumption that the attacker knows the multi-channel model architecture and used defense strateg, but does not know the defenders' secret keys, and (iii) black-box direct attacks under assumption that the attacker does not know the model architecture or defense mechanisms. In all scenarios, the worst case assumption is that the attacker uses the same data set as the defender. Although the numbers of classification accuracy and classification errors deviate from those that the original authors have, we anyway confirmed the robustness of the method, although some results are not universal.

8. References

[1] O. Taran, S. Rezaeifar, T. Holotyak, S. Voloshynovskiy. "Machine Learning Through Cryptographic Glasses: Combating Adversarial Attacks by Key-Based Diversified Aggregation." (2020). Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7357678/pdf/13635_2020_Article_106.pdf

[2] O. Taran, S. Rezaeifar, T. Holotyak, S. Voloshynovskiy, "Defending Against Adversarial Attacks by Randomized Diversification." (2019). Available: https://arxiv.org/pdf/1904.00689.pdf

[3] O. Taran, S. Rezaeifar, T. Holotyak, S. Voloshynovskiy, "Robustification of Deep Net Classifiers by Key Based Diversified Aggregation with Pre-filtering." (2019). Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8803714

[4] O. Taran, S. Rezaeifar, S. Voloshynovskiy, "Bridging Machine Learning and Cryptography in Defense Against Adversarial Attacks." (2018). Available: https://arxiv.org/pdf/1809.01715.pdf

[5] N. Carlini, D. Wagner, "Towards Evaluating the Robustness of Neural Networks." (2017). Available: https://arxiv.org/pdf/1608.04644.pdf

[6] J. Su, D. V. Vargas, S. Kouichi, "One Pixel Attack for Fooling Deep Neural Networks." (2017). Available: https://arxiv.org/pdf/1710.08864.pdf

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks." (2017). Available: https://arxiv.org/pdf/1706.06083.pdf

[8] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and Harnessing Adversarial Examples." (2014). Available: https://arxiv.org/pdf/1412.6572.pdf

[9] R. Storn, K. Price. "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces." Journal of Global Optimization 11, 341–359 (1997). https://doi.org/10.1023/A:1008202821328.

[10] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian. "Image denoising by sparse 3-d transform-domain collaborative filtering." (2008). Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4271520

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol. "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion." (2010). Available: https://dl.acm.org/doi/pdf/10.5555/1756006.1953039

[12] A. Athalye, N. Carlini, D. Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples." (2018). Available: http://proceedings.mlr.press/v80/athalye18a/athalye18a.pdf

[13] K. Simonyan, A. Zisserman. "Very deep convolutional networks for large-scale image recognition." (2014). Available: https://arxiv.org/pdf/1409.1556.pdf

[14] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition." (2016). Available: https://arxiv.org/pdf/1512.03385.pdf