Towards robust stacked capsule autoencoder with hybrid adversarial training

Jiazhu Dai · Siwei Xiong

Accepted: 5 September 2023 / Published online: 22 September 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
Capsule networks (CapsNets) are new neural networks that classify images based on the spatial relationships of features. By analyzing the pose of features and their relative positions, it is more capable of recognizing images after affine transformation. The stacked capsule autoencoder (SCAE) is a state-of-the-art CapsNet that achieved unsupervised classification of CapsNets for the first time. However, the security vulnerabilities and the robustness of the SCAE have rarely been explored. In this paper, we propose an evasion attack against SCAE, where the attacker can generate adversarial perturbations by reducing the contribution of the object capsules related to the original category of the image in the SCAE. Adversarial perturbations are then applied to the original images, and the perturbed images are misclassified with a high probability. For such an evasion attack, we further propose a defense method called hybrid adversarial training (HAT), which makes use of adversarial training and adversarial distillation to achieve better robustness of SCAE against the evasion attack. We evaluate the defense method and the experimental results show that the SCAE trained with HAT ensures that the model can maintain relatively high classification accuracy under the evasion attack and achieve similar classification accuracy to that of the original SCAE model on clean samples. The source code is available at https://github.com/FrostbiteXSW/SCAE_Defense.

Keywords Stacked capsule autoencoder · Evasion attack · Evasion attack defense · Image classification

1 Introduction
Convolutional neural networks (CNNs) perform well when handling computer vision tasks. Although CNNs have excellent fitting and generalization capabilities, and can obtain accurate recognition results among various datasets and samples, CNNs are not good at addressing affine transformations such as rotation and resizing.

The original intention of capsule networks (CapsNets) is to solve the problem of CNNs when dealing with affine transformations. CapsNets extract parts of different objects in the image and analyze their relationships to reveal the object compositions and their correlations. The stacked capsule autoencoder (SCAE) [1] is a state-of-the-art CapsNet that introduces the theory of CapsNets into autoencoders. By capturing the pose, presence and features of different parts and objects, the SCAE can conduct unsupervised classification on images.

However, recent studies have proven that CapsNets are vulnerable to adversarial attacks [2–5], as is the SCAE. In this paper, we propose an evasion attack against the SCAE. After a perturbation is generated based on the output of the SCAE’s object capsules to decrease their contribution to correct classification, it is applied to the clean image to form an adversarial image. Then the adversarial image is input into the SCAE and misclassified by the classifiers of the SCAE. According to our studies and experiments, it is possible to construct imperceptible adversarial samples to deceive the SCAE, which may cause its predictions to no longer be reliable and raise concerns for applying it in safety-critical applications [6].

At the moment, there are studies aiming at improving the resistance of CapsNets to adversarial attacks [7–9], but these studies focus on CapsNets with dynamic routing, and the adversarial robustness of the SCAE has rarely been explored. In this paper, we propose a defense method called hybrid...
adversarial training (HAT) against the above threat, which is based on adversarial training and adversarial distillation. Adversarial training helps the SCAE to discover and fix vulnerabilities by inserting adversarial samples into training datasets. Adversarial distillation ensures the classification accuracy of the SCAE on clean samples. HAT combines the above two defense methods, and not only ensures accuracy but also further improves the robustness of the SCAE. The experimental results prove that our defense method can enhance the resistance of the SCAE to adversarial attacks and achieve 82.14% classification accuracy under evasion attacks. The contributions of the paper can be summarized as follows:

1. We propose an evasion attack method against the SCAE, which can cause the classifiers of the SCAE to output incorrect predictions. The evasion attack method confirms the existence of security vulnerabilities in the SCAE, and it has high attack success rate and stealthiness;
2. We propose a defense method called HAT against the above evasion attack, where neither the original structure of the SCAE is modified, nor new modules are added during the testing phase. Our defense method ensures that the SCAE can maintain relatively high classification accuracy on adversarial samples and achieve similar classification accuracy to that of the original model on clean samples.

The remainder of this paper is organized as follows. Section 2 introduces the related works. Section 3 explains the architecture and operations of the SCAE, and the theory of adversarial training and distillation. Section 4 describes the threat model. Section 5 presents our defense method in detail. Section 6 describes the experiments and results of our defense method. Section 7 provides a summary and briefly presents our future work.

2 Related works

2.1 Capsule networks

CapsNets are a type of model that recognize images according to spatial relationships. To date, CapsNets have developed three different versions: the dynamic routing capsule network proposed by Sabour et al. [10] in 2017, the EM routing capsule network proposed by Hinton et al. [11] in 2018, and the stacked capsule autoencoder proposed by Kosiorek et al. [1] in 2019.

As a state-of-the-art CapsNet, the SCAE is different from past CapsNets in that it focuses on unsupervised learning and has autoencoders but no routing algorithm in its structure. The SCAE model consists of the part capsule autoencoder (PCAE) and the object capsule autoencoder (OCAE). The PCAE decomposes the input image into small parts, and passes the information of these parts to the OCAE. The OCAE composes the parts into different objects, and outputs the presence probability of all known objects. Finally, a classifier categorizes the image based on the objects that appear in it.

The PCAE and the OCAE are composed of special neurons called capsules. A part capsule inside the PCAE contains a six-dimensional pose vector, a one-dimensional presence probability and an n-dimensional attribute vector. The object capsules inside the OCAE are encoded by a set transformer [12], each of which is composed of multiple part capsules. For the classifiers, the SCAE uses k-means classifiers for unsupervised classification, and provides optional linear classifiers for supervised classification.

The main contribution of the SCAE is a new method for image classification, that is, to decompose the image into several small parts and recompose them into larger objects. During this procedure, the SCAE obtains the relationships between different parts and objects. Unlike CNNs using local features to classify images, the SCAE considers the spatial relationships between features and the variety of representations of similar features to suppress the influence of the change in a single feature, which gives the SCAE better resistance to random perturbations. The SCAE can achieve 98.7% accuracy of unsupervised classification on the MNIST dataset.

2.2 Adversarial attacks

Adversarial attacks against machine learning can be categorized into two types: poisoning attacks and evasion attacks. Poisoning attacks, which occur during the training phase, aim at degrading the performance of the model or creating backdoors in the model to control its behavior. The attacker controls the training process of the model to create a backdoor inside it by adding elaborately constructed malicious samples to the training dataset to make the model output the results specified by the attacker on the samples containing the backdoor pattern, or decrease the model accuracy in the testing phase [13–22].

Evasion attacks, which occur during the testing phase, aim at creating adversarial samples to deceive the model. The attacker applies an imperceptible perturbation to the clean sample to form an adversarial sample. The adversarial sample will be misclassified or categorized as the attacker specifies by the model [23–33]. The defense methods proposed in this paper target evasion attacks.
2.3 Adversarial attack defenses

The defenses against adversarial attacks for machine learning models are categorized into the following three broad categories [34]:

1. **Gradient Masking/Obfuscation**: This method studies how to hide the gradient information of the model to confuse adversaries, since most attack algorithms are based on the gradient information of the model [35];

2. **Robust Optimization**: Robust optimization aims to improve the robustness of the models by changing the model parameters with relearning so that the trained model will correctly classify the subsequent adversarial examples [7–9, 23, 36–48];

3. **Adversarial Examples Detection**: Adversarial example detection is another main approach to protect the models. It detects the presence of adversarial perturbations in the input during inference, and if the input is adversarial, it disallows the input into the models [49–57].

Adversarial training and adversarial distillation, which both belong to robust optimization, are widely used to defend against adversarial attacks.

The main objective of adversarial training is to make a model more robust by training it with a dataset containing benign and adversarial samples. Adversarial training is effective for defending against specific adversarial attacks, but the training procedure is complex and time-consuming.

Adversarial distillation is an improvement on adversarial training that introduces distillation into the training procedure. With the help of distillation, the model can achieve better robustness, but the training procedure is more complex than adversarial training.

Our proposed defense method belongs to robust optimization.

2.4 Security issues about capsule networks

After the emergence of CapsNets, research on their robustness and security application mainly focuses on the dynamic routing capsule network. Frosst et al. [49] detected adversarial samples by measuring the Euclidean distances between input samples and their reconstructions generated by the dynamic routing capsule network. Qin et al. [50] constructed a network with cycle-consistent winning capsule reconstructions based on the dynamic routing capsule network, and used three detectors to filter adversarial samples. Deng et al. [51] filtered adversarial samples using a CapsNet with refined dynamic routing algorithm.

However, CapsNets also face security threats. Yoon [2] transferred several adversarial attacks to the dynamic routing capsule network and successfully fooled it. Michels et al. [3] proved that the dynamic routing capsule network is not more resistant to white-box adversarial attacks than CNNs. Marchisio et al. [4] designed a black-box adversarial attack against the dynamic routing capsule network and verified its effectiveness on the GTSRB dataset. De Marco [5] proved that the dynamic routing capsule network is vulnerable to adversarial attacks.

As the security of the dynamic routing capsule is questionable, its security applications are no longer reliable. Li et al. [7] improved the robustness of the dynamic routing capsule network by adversarial training. Peer et al. [8] designed a capsule network with \( \gamma \)-capsules by adding a new inductive bias and replacing the routing algorithm, which is more capable of defending against adversarial attacks. Garg et al. [9] enhanced the adversarial robustness of the dynamic routing capsule network via thermometer encoding and adversarial training.

The above studies are limited to CapsNets with routing algorithms. Although SCAPE is a state-of-the-art CapsNet, few studies have explored the security of SCAPE. In this paper, we confirm the security vulnerabilities of the SCAPE by an evasion attack, that is, adding perturbations to input samples to induce the output of the SCAPE to cause misclassification [6]. Moreover, we propose a defense method called hybrid adversarial training (HAT) against the above evasion attack of the SCAPE.

3 Preliminaries

3.1 Stacked capsule autoencoder

Figure 1 shows the main structure of the SCAPE. The SCAPE recognizes images based on two core modules: the part capsule autoencoder (PCAE) and the object capsule autoencoder (OCAE). First, the PCAE uses a CNN to extract the pose, presence and features of each part of the objects in the input image. Then, the OCAE uses a set transformer [12] to encode the scattered parts obtained by the PCAE into larger objects, and outputs the presence probability of all objects that may appear. Finally, a classification result is given by the classifier according to the output of the OCAE.

There are two types of outputs provided to the classifiers by the SCAPE: the prior object capsule presence with dimension \([B, K]\) and the posterior object capsule presence with dimension \([B, K, M]\), where \(B\) is the batch size, \(K\) is the number of object capsules, and \(M\) is the number of part capsules. The SCAPE tells the classifiers the presence probability through these outputs, whose values range from 0 to 1.

A classifier collects the outputs of the SCAPE on the training dataset and chooses one type of output for k-means clustering. Then it uses bipartite graph matching [58] to obtain the mapping permutation between clustering labels
and ground truth labels. In the testing phase, the trained k-means classifier receives the output of the SCAE and predicts the label of the input image. It is noticeable that to make different k-means classifiers have the same structure, the dimension $M$ of the posterior object-capsule presence is reduced and summed to ensure shape consistency.

### 3.2 Adversarial training

Adversarial training is a training technique that improves the robustness of the model. In this theory, adversarial samples that can fool the model are added into the training dataset and involved in the training procedure. The model can learn the information of its vulnerabilities through these samples and enhance the ability to resist adversarial attacks. Common adversarial training methods add the distance between the outputs of the model on the adversarial samples and their corresponding clean samples to the loss function during the training procedure so that the trained model can obtain the right outputs on adversarial samples at the testing phase. Formula (1) shows the modified loss function:

$$\tilde{L}(x, \bar{y}, M) = \|\bar{y} - M(x')\|_2$$

where $x$ is the clean sample, $\bar{y}$ is the one-hot encoding of the label of $x$, $x'$ is the adversarial sample that can fool the model $M$ based on $x$, and $M(x')$ denotes the output of $M$ on $x'$. Effective adversarial training is required to improve the robustness of the model and reduce the impact on the accuracy of the model as much as possible. However, the complexity of the algorithm to generate the adversarial samples for training should be limited to reduce the computation time. At present, there are studies of adversarial training such as [23, 36–39].

### 3.3 Distillation

Proposed by Hinton et al. [59] in 2015, distillation is a technique that compresses the model while maintaining its accuracy. Usually, the more complex the model is, the stronger its fitting ability is, but the cost is that the model may become too large to be deployed on terminals. The theory of distillation is to train a complex teacher model $M_{tch}$ and then train a simplified student model $M_{stu}$ with the guidance of $M_{tch}$. The knowledge of $M_{tch}$ is transferred to $M_{stu}$, and the two models can have similar accuracy. The key issue of distillation is the method of knowledge transfer. Hinton et al. modified the softmax function:

$$\tilde{F}(x) = \left[ \frac{e^{z_i(x)}}{\sum_{j=0}^{N-1} e^{z_j(x)}} \right]_{i \in 0..N-1}$$

where $x$ is the input sample, $z(x)$ is the logits output by the last hidden layer of the model, $N$ is the length of the logit vector, and $T$ is a temperature hyper-parameter. This function is called softmax with $T$. The larger the value of $T$ is, the smoother the output of the function, and the less difficult it is to fit it as the target. During the training phase, the original softmax functions of $M_{stu}$ and $M_{tch}$ are replaced by Formula (2), and the distance of the outputs of the two models on the same sample is added to the loss function of $M_{stu}$. Through this approach, the knowledge of $M_{tch}$ can be transferred to $M_{stu}$.

Adversarial distillation is a special distillation use. During the distillation procedure, not only is the knowledge of $M_{tch}$ learned by $M_{stu}$ but the robustness of $M_{tch}$ is also inherited by $M_{stu}$. If $M_{stu}$ relies only on itself to improve its robustness (for example, using adversarial training), the features of adversarial samples may not be recognized by $M_{stu}$ well, or $M_{stu}$ may be overfitted on adversarial samples. Nevertheless, these problems can be mitigated by distillation. For example, combining adversarial training and distillation or distilling $M_{stu}$ with robust $M_{tch}$ are both effective methods to train robust models [46–48].

### 4 Threat model

In this section, we propose an evasion attack against the SCAE. The attacker constructs an adversarial perturbation and applies it to the clean sample to form an adversarial sample. The adversarial sample induces the encoding process of
the SCAE by decreasing the presence value which is output by the object capsules related to the clean sample label to lower the contribution of these capsules to correct classification. The SCAE outputs incorrect encoding results on the adversarial sample, and the classifier makes incorrect classifications based on the SCAE’s output.

The SCAE uses the sparsity loss to allocate different capsules to different classes, and the object capsules related to the correct class are activated and high values are output, while the irrelevant object capsules remain inactive. The evasion attack aims at lowering the output of those activated object capsules to cause the SCAE to mistakenly believe that “the objects that belong to the real category do not appear in the image”. This theory can be described as the following target function:

\[
\text{Minimize } f (x + p) = \sum_{i \in S} E (x + p)_i
\]  

(3)

where \(x\) is the clean sample, \(p\) is the adversarial perturbation, \(E (x)_i\) is the value of presence output by the object capsule \(i\) in the SCAE model \(E\) on the input \(x\), and \(S\) is the object capsule subset related to the original category of \(x\):

\[
S = \{ i \mid E (x)_i > \frac{1}{K} \sum_j E (x)_j \}
\]

(4)

The subset \(S\) includes all object capsules whose outputs are greater than the average value. According to the target function of Formula (3), we can obtain the optimization problem of the evasion attack:

\[
\text{Minimize } \|p\|_2 + \alpha \cdot f (x + p)
\]

\[s.t. \ x + p \in [0, 1]^n\]

(5)

where \(\alpha > 0\) is a suitably chosen hyperparameter that ensures that the two parts of Formula (5) can be optimized simultaneously. For the box constraints in Formula (5), namely, \(x + p \in [0, 1]^n\), to avoid the damage of gradient propagation caused by clipping pixel values directly, we use the "change of variables" method to handle it. We map the original sample into the \(\text{arctanh}\) space, compute it on the mapped sample, and finally, map it back to the \([0, 1]^n\) space. The relationship between the original and mapped samples is as follows:

\[
w = \text{arctanh} (2x - 1)
\]

\[
p = \frac{1}{2} \left( \text{tanh} (w + p) + 1 \right) - x
\]

(6)

where \(w\) is the mapping of the original sample \(x\) in the \(\text{arctanh}\) space, and \(p\) is the mapping of the perturbation \(p\) in the \(\text{arctanh}\) space. Our algorithm computes \(p\) in the \(\text{arctanh}\) space instead of \(p\). The range in values does not need to be considered because the hyperbolic tangent function can map any value in \((-\infty, +\infty)\) back to \((-1, 1)\).\(^1\)

The full algorithm consists of the inner iteration and the outer iteration. In the inner iteration, we use an optimizer to solve the optimization problem of Formula (5). In the outer iteration, we initialize the optimizer, execute a complete inner iteration and update the value of \(\alpha\). We perform multiple rounds of outer iterations and select the perturbation \(p\) with the smallest \(\|p\|_2\) as the best result. The whole procedure is shown in Algorithm 1.

---

**Algorithm 1 Generating the perturbation with optimizer.**

1: **Input:** Image \(x\), SCAE model \(E\), classifier \(C\), optimizer \(opt\), hyperparameter \(\alpha\), the number of outer iterations \(n_{\text{outer}}\), the number of inner iterations \(n_{\text{inner}}\).

2: **Output:** Perturbation \(p\).

3: 4: Initialize \(p \leftarrow 0\), \(L_p \leftarrow +\infty\).

5: 6: \(w \leftarrow \text{arctanh} (2x - 1)\)

7: 8: for \(i \in n_{\text{inner}}\) do

9: 10: for \(j \in n_{\text{inner}}\) do

11: 12: \(x_j^{\text{adv}} \leftarrow \frac{1}{2} \left( \text{tanh} (w + p_j) + 1 \right)\)

13: 14: \(E \leftarrow \|x_j^{\text{adv}} - x\|_2 + \alpha \cdot \sum_{k \in S} E (x_k^{\text{adv}})\)

15: 16: \(p_{j+1} \leftarrow \text{opt} \left( p_j \mid E \right)\)

17: 18: if \(C (E (x^{\text{adv}}_{j+1})) = C (E (x))\) and \(\|x_j^{\text{adv}} - x\|_2 < L_p\) then

19: 20: \(L_p \leftarrow \|x_j^{\text{adv}} - x\|_2\)

21: end if

22: end for

23: Update \(\alpha\)

24: end for

---

We set the values of the upper bound \(\alpha_{ub}\) and lower bound \(\alpha_{lb}\) of \(\alpha\). During the update of \(\alpha\), if the algorithm obtains any adversarial sample \(x^{\text{adv}}_{j+1}\) that satisfies \(C (E (x^{\text{adv}}_{j+1})) \neq C (E (x))\) in the inner iterations, we let \(\alpha_{ub} \leftarrow \alpha\), otherwise \(\alpha_{lb} \leftarrow \alpha\), and finally, take \((\alpha_{ub} + \alpha_{lb}) / 2\) as the new value of \(\alpha\) in the next outer iteration.

We conducted experiments on multiple datasets to verify the effectiveness of our evasion attack, and the details can be found in [6]. The experimental results show that our evasion attack can achieve high attack success rates and stealthiness,

---

\(^1\) During the experiments, we use arctanh \((2x - 1) + \epsilon\) to avoid dividing by zero.
which proves the existence of security vulnerabilities in the SCAE.

5 The proposed method to train robust SCAE

In this section, we propose a defense method called hybrid adversarial training (HAT), and the goal of HAT is to improve the robustness of the SCAE against the evasion attack while maintaining its classification accuracy as much as possible. HAT is a combination of adversarial training and adversarial distillation, which are widely used to defend against adversarial attacks on neural network models. The defense models of SCAE with adversarial training and adversarial distillation are presented in Sections 5.1 and 5.2 respectively. Then the details of HAT are described in Section 5.3.

5.1 Training SCAE with adversarial training

We name the defense model with adversarial training AT-SCAE. In this method, we generate adversarial samples that can cause misclassification (see Appendix A for details) and add the generated adversarial samples to the training dataset so that the SCAE can adjust its parameters according to the features of the adversarial samples during the training phase.

The adversarial training method described in Formula (1) in Section 3.2 cannot be applied to the SCAE directly because the training procedure of the SCAE is unsupervised and the labels of the samples are unavailable. Additionally, the outputs encoded by the SCAE are the presence probabilities of different objects, and the classification results cannot be obtained until the encoding results are input into the k-means classifier.

To conduct adversarial training on the SCAE, we need to make use of the following mechanism in the unsupervised training.

When the SCAE encodes an image into capsules, it also decodes the capsules back into the image to ensure that the capsules contain valid information about the image. This step is optimized by the reconstruction loss, which is the distance between the decoded image and the input image [1]. When conducting adversarial training, we modify the computation of the reconstruction loss. If the input image is an adversarial sample, the reconstruction loss will be the distance between the decoded image and the corresponding clean sample of the adversarial sample. During the procedure of encoding and decoding, the SCAE discards the adversarial features and only retains the clean features. The loss function of AT-SCAE is as follows:

$$ L_{AT}(x, x', E) = L_{no_{rec}}(x, E) + L_{rec}(x, x', E) $$ (7)

where $E$ is the SCAE model, $x$ is the input sample, $L_{no_{rec}}(x, E)$ represents all loss items in the loss function of the SCAE except for the reconstruction loss, and $L_{rec}(x, x', E)$ represents the reconstruction loss with the target image $x'$.

We take the input sample $x$ as the target image $x'$ of the reconstruction loss for normal training. After $k$ batches of normal training, we conduct one batch of adversarial training; that is, we take the generated adversarial samples according to Appendix A as the input samples $x$ and their corresponding clean samples as the target images $x'$ of the reconstruction loss, whereby we train the SCAE with the loss function of Formula (7) to fix potential vulnerabilities. The reason why this procedure is conducted at intervals is to mitigate the impact on the accuracy of the SCAE. The whole training procedure of AT-SCAE is shown in Algorithm 2.

Algorithm 2 Training the AT-SCAE model.

1: Input: Training dataset $X$, optimizer $opt$, generator for adversarial samples $G$, the number of epochs $n_{ep}$, the number of batches before conducting one batch of adversarial training $k$.  
2: Output: AT-SCAE model $E_{AT}$. 
3: 4: Initialize AT-SCAE model $E_{AT}$ with parameters $\theta$, $n_{bch} \leftarrow 0$.  
5: for epoch in $n_{ep}$ do  
6: for each batch $x$ in $X$ do  
7: if $n_{bch} = k$ then  
8: /* Conduct one batch of adversarial training */  
9: $\theta \leftarrow opt(\theta | L_{AT}(G(x), x, E_{AT}))$  
10: $n_{bch} \leftarrow 0$  
11: else  
12: /* Conduct $k$ batches of normal training */  
13: $\theta \leftarrow opt(\theta | L_{AT}(x, x, E_{AT}))$  
14: $n_{bch} \leftarrow n_{bch} + 1$  
15: end if  
16: end for  
17: end for  
18: return $E_{AT}$

5.2 Training SCAE with adversarial distillation

We name the defense model with adversarial distillation AD-SCAE. This method introduces the theory of distillation to improve the accuracy of the robust SCAE model. Although AT-SCAE can enhance the robustness of the SCAE, it decreases the accuracy of the SCAE to a certain degree. In the training procedure of AD-SCAE, we obtain a teacher model with normal training and then train a student model with the help of the teacher model based on adversarial distillation. The teacher model guides the student model to correctly distinguish the characteristics of the adversarial samples to reduce their impact on the accuracy of the student model.
We conduct adversarial distillation on the output of the capsules encoded by the SCAE. Different from the traditional distillation process, AD-SCAE adds the distance between the outputs of the teacher model on the original sample and the outputs of the student model on the adversarial sample to the loss function as the distillation loss. The knowledge of the teacher model is transferred to the student model and guides the student model to fix its vulnerabilities according to the features of the adversarial samples. As the output layer of the SCAE does not have a softmax function, there is no need to process the output with Formula (2). The modified loss function is as follows:

\[
L_{AD}(x, x', E_{ich}, E_{stu}, \lambda) = (1 - \lambda) * L_{AT}(x, x', E_{stu}) + \lambda * \|E_{ich}(x) - E_{stu}(x')\|_2
\]  

(8)

where \(x\) is the original sample, \(x'\) is the input sample for the student model (it can be the original sample or the adversarial sample), \(E(x)\) represents the encoding result of the SCAE model \(E\) on the input sample \(x\), \(E_{ich}\) and \(E_{stu}\) represent the teacher model and the student model with the same structure as the SCAE, and \(\lambda\) is the weight that balances the two loss items.

We conduct one batch of adversarial distillation every \(k\) batches of normal distillation. Both normal and adversarial distillation use the loss function of Formula (8) to train the SCAE. During normal distillation, \(x'\) is the same as the original sample \(x\). When conducting adversarial distillation, we take the generated adversarial samples according to Appendix A as \(x'\). The training procedure of AD-SCAE is shown in Algorithm 3.

### 5.3 Training SCAE with HAT (Hybrid Adversarial Training)

We name the defense model with hybrid adversarial training HAT-SCAE. This method is a combination of adversarial training and adversarial distillation that aims to obtain better accuracy and robustness. AD-SCAE promotes the SCAE’s accuracy, but it has some negative effects on the SCAE’s robustness. According to the research of Zhu et al. [48], if the robustness of the teacher model for distillation is unreliable, this unreliability can be transferred to the student model and cause it to have the same vulnerabilities as the teacher model, which is also confirmed by our experimental results. To solve this problem, we propose the hybrid adversarial training method in this section.

Adversarial distillation can promote model accuracy and improve its robustness to a certain degree, while adversarial training focuses on improving the robustness of the model.

Hybrid adversarial training combines adversarial distillation and adversarial training to train a more robust SCAE model. First, we use the loss function of AD-SCAE \(L_{AD}\) to train the SCAE model and ensure that it has enough knowledge of the clean samples and can distinguish adversarial samples to a certain degree. After the SCAE model has enough accuracy and no longer needs the guidance of a teacher model, we use the loss function of AT-SCAE \(L_{AT}\) to train the SCAE model to enhance its robustness and fix the vulnerabilities inherited from the teacher model. The HAT-SCAE’s training procedure is shown in Algorithm 4.

### 6 Experimental evaluation

In the experiments, we evaluate the effectiveness of the defense method HAT against the evasion attack. The experiments consist of two parts: one evaluates the impact of HAT on the classification performance of the SCAE on clean samples, and the other evaluates the robustness of the HAT against the evasion attack on adversarial samples. The source code of our experiments is available at [https://github.com/FrostbiteXSW/SCAE_Defense](https://github.com/FrostbiteXSW/SCAE_Defense).

#### 6.1 Experimental setup

**Baseline** As there exist few studies dedicated to improving the security robustness of the SCAE, and as both adversarial training and adversarial distillation are widely used to defend against adversarial attacks on neural network models, we use them as the baseline to evaluate the performance of
Algorithm 4 Training the HAT-SCAE model.

1: **Input:** Training dataset $X$, optimizer $\text{opt}$, generator for adversarial samples $G$, pretrained teacher model $E_{\text{ICH}}$, hyperparameter $\lambda$, the number of epochs for adversarial distillation $n_{\text{ad}}$, the number of epochs for adversarial training $n_{\text{tr}}$, the number of batches before conducting one batch of adversarial distillation or adversarial training $k$.
2: **Output:** HAT-SCAE model $E_{\text{HAT}}$.
3: 4: Initialize HAT-SCAE model $E_{\text{HAT}}$ with parameters $\theta$, $n_{\text{bch}} \leftarrow 0$.
5: /* Phase 1: Adversarial Distillation */
6: for epoch in $n_{\text{ad}}$ do
7:   for each batch $x$ in $X$ do
8:     if $n_{\text{bch}} = k$ then
9:       /* Conduct one batch of adversarial distillation */
10:      $\theta \leftarrow \text{opt}(\theta | \mathcal{L}_D(x, G(x), E_{\text{ICH}}, E_{\text{HAT}}, \lambda))$
11:      $n_{\text{bch}} \leftarrow 0$
12:     else
13:       /* Conduct $k$ batches of normal distillation */
14:      $\theta \leftarrow \text{opt}(\theta | \mathcal{L}_D(x, x, E_{\text{ICH}}, E_{\text{HAT}}, \lambda))$
15:      $n_{\text{bch}} \leftarrow n_{\text{bch}} + 1$
16:   end if
17: end for
18: end for
19: /* Phase 2: Adversarial Training */
20: for epoch in $n_{\text{tr}}$ do
21:   for each batch $x$ in $X$ do
22:     if $n_{\text{bch}} = k$ then
23:       /* Conduct one batch of adversarial training */
24:      $\theta \leftarrow \text{opt}(\theta | \mathcal{L}_A(x, G(x), x, E_{\text{HAT}}))$
25:      $n_{\text{bch}} \leftarrow 0$
26:     else
27:       /* Conduct $k$ batches of normal training */
28:      $\theta \leftarrow \text{opt}(\theta | \mathcal{L}_A(x, x, E_{\text{HAT}}))$
29:      $n_{\text{bch}} \leftarrow n_{\text{bch}} + 1$
30:   end if
31: end for
32: end for
33: return $E_{\text{HAT}}$

HAT against the evasion attack. We train the defense models of SCAE (i.e., the AT-SCAE, the AD-SCAE and the HAT-SCAE) with adversarial training, adversarial distillation and HAT separately and compare their classification accuracy (the ratio of the number of samples that are classified correctly by the classifier to the total number of all test samples) on clean and adversarial samples.

Datasets Because distillation has certain requirements for the accuracy of the teacher model, when applying the defense methods to datasets that are difficult to identify by the SCAE, the improvement of robustness will be limited. The experimental results given by Kosiorek et al. in the original paper on the SCAE [1] show that although the SCAE can achieve 98.7% accuracy on the MNIST dataset, it can only achieve 55.33% and 25.01% accuracy on the SVHN and CIFAR10 datasets, which does not satisfy the need of our experiments. ImageNet, which is similar to CIFAR10, is also not suitable for our experiments. After comparing the performance of the SCAE on various datasets, we select the MNIST and Fashion-MNIST datasets for experiments to show the influence of different defense methods on the robustness of the SCAE.

Parameters We train the SCAE models on the two datasets with the same parameters as those of the original SCAE [1, 60], which are shown in Table 1. For part CNN, $2 \times (128:2)$ denotes two convolutional layers with 128 channels and a stride of two. For the set transformer, $3 \times (1-16)-256$ denotes three layers, one attention head, 16 hidden units and 256 output units. All models use the same optimizer for training. The main parameters of the optimizer are shown in Table 2.

According to the information given by Kosiorek et al. [1], the values of $k$ of all k-means classifiers are set as 10, which is the same as the number of categories of the datasets.

Before the defense experiments, the teacher models for distillation need to be pretrained. The accuracy of all teacher models is shown in Table 3.

| Dataset                  | MNIST | Fashion-MNIST |
|--------------------------|-------|---------------|
| Canvas size              | 40    | 40            |
| Num of part capsules     | 24    | 24            |
| Num of object capsules   | 24    | 24            |
| Num of channels          | 1     | 1             |
| Template size            | 11    | 11            |
| Part capsule noise scale | 4.0   | 4.0           |
| Object capsule noise scale| 4.0  | 4.0           |
| Part CNN                 | 2x(128:2)-2x(128:1) | 2x(128:2)-2x(128:1) |
| Set transformer          | 3x(1-16)-256 | 3x(1-16)-256 |

Table 2 Settings of the optimizer to train the SCAEs

| Optimizer parameter | Value       |
|---------------------|-------------|
| Algorithm            | RMSProp     |
| Learning rate        | $3 \times 10^{-5}$ |
| Momentum             | 0.9         |
| $\epsilon$           | $1 \times 10^{-6}$ |
| Learning rate decay steps | 10000 |
| Learning rate decay rate | 0.96 |
| Batch size           | 100         |
6.2 Experimental method

Our experiments consist of two parts: one evaluates the influence of the HAT on the classification performance of the SCAE by comparing the classification accuracy of the original SCAE model, the baseline models (i.e., the AT-SCAE and the AD-SCAE) and our defense model HAT-SCAE on clean samples; the other evaluates the robustness of the defense method HAT against the evasion attack by comparing the classification accuracy of the original SCAE model, the baseline models (i.e., the AT-SCAE and the AD-SCAE) and our defense model HAT-SCAE on adversarial samples.

The classification accuracy mentioned above refers to the ratio of the number of samples that are classified correctly by the classifiers of the models to the total number of all test samples. Our experiments consist of two steps:

1. We train the original SCAE, the AT-SCAE, the AD-SCAE and the HAT-SCAE on the MNIST and Fashion-MNIST datasets, construct the prior k-means classifiers and the posterior k-means classifiers for each model separately, and finally, evaluate the classification accuracy of the models on clean samples, which is shown in Table 5.

2. We attack the prior k-means classifiers and the posterior k-means classifiers on the MNIST and Fashion-MNIST datasets with the evasion attack method proposed in Section 4. In each attack experiment, we randomly choose 5,000 samples and generate adversarial samples based on them. Then we input the adversarial samples to the original SCAE, the AT-SCAE, the AD-SCAE and the HAT-SCAE to obtain the encoding results of the adversarial samples, which are then input into the classifiers to obtain the classification results. We evaluate the classification accuracy of the models on adversarial samples, which is shown in Table 6.

In the experiments, the number of epochs of training \(n_{\text{ep}}\) in AT-SCAE and AD-SCAE is set to 100. The number of epochs of adversarial distillation \(n_{\text{ad}}\) and the number of epochs of adversarial training \(n_{\text{at}}\) in HAT-SCAE are set to 50. The number of interval batches between two batches of adversarial training and adversarial distillation \(k\) in all defense methods is set to 1. The weight value \(\lambda\) in Formula (8) is set to 0.5.

For the adversarial sample generation algorithm for training, we set the numbers of inner and outer iterations to 5 and 30 respectively. The hyperparameter \(\beta\) in Formula (A1) is set to 1. The initial value of hyperparameter \(\alpha\) is set to 100, and the upper and lower bounds of \(\alpha\) are set to \(+\infty\) and 0. If the upper bound is \(+\infty\) when updating \(\alpha\), let \(\alpha \leftarrow \alpha \ast 10\). The target classifier of the generation algorithm of adversarial samples for training is the built-in posterior linear classifier of the SCAE [1] instead of the k-means classifiers, and the reason for this is to remove the time cost of updating the k-means classifiers repeatedly to accelerate the training procedure.

For the evasion attack algorithm for robustness evaluation, we set the numbers of inner and outer iterations to 9 and 300 respectively. The initial value of hyperparameter \(\alpha\) is set to 100, and the upper and lower bounds of \(\alpha\) are set to \(+\infty\) and 0. If the upper bound is \(+\infty\) when updating \(\alpha\), let \(\alpha \leftarrow \alpha \ast 10\). The main parameters of the optimizer used by the evasion attack algorithm are shown in Table 4.

The attack success rate increases with the amount of perturbation, while the stealthiness worsens. The amount of perturbation needs to be limited to objectively evaluate the effect of different defense methods. In our experiments, the thresholds of the \(L_2\) norms of the perturbations on the MNIST and Fashion-MNIST datasets are set as 4 and 5, respectively, which ensure that the adversarial perturbations are imperceptible.

6.3 Results and discussion

The comparison of classification accuracy of the original SCAE and the three defense models (i.e., the AT-SCAE, the AD-SCAE and the HAT-SCAE) on the clean sample is shown in Table 5. The bold numbers in the table represent the best performance.

From the table, we have the following results: for clean samples from the MNIST dataset, when using either the prior k-means classifier or the posterior k-means classifier, the classification accuracy of the HAT-SCAE is better than that of the AT-SCAE, and it is very close to those of the original SCAE and the AD-SCAE, which have the best performance.

Table 3 Accuracy of the pretrained teacher models

| Dataset               | MNIST | Fashion-MNIST |
|-----------------------|-------|---------------|
| Prior k-means classifier | 97.40 | 67.71         |
| Posterior k-means classifier | 97.63 | 66.83         |

Table 4 Settings of the optimizer used to attack the SCAEs

| Optimizer parameter | Value             |
|---------------------|-------------------|
| Algorithm           | Adam              |
| Learning rate       | 1.0               |
| \(\rho_1\)          | 0.9               |
| \(\rho_2\)          | 0.999             |
| \(\epsilon\)        | 1x10\(^{-8}\)     |
The comparison of classification accuracy of the original SCAE and the three defense models on clean samples

| Classifier       | Model   | MNIST (%) | Fashion-MNIST (%) |
|------------------|---------|-----------|-------------------|
| Prior k-means classifier | SCAE    | 97.40     | 67.71             |
|                  | AT-SCAE | 85.18     | 58.87             |
|                  | AD-SCAE | 97.68     | 61.15             |
|                  | HAT-SCAE| 95.22     | 59.17             |
| Posterior k-means classifier | SCAE   | 97.63     | 66.83             |
|                  | AT-SCAE | 80.97     | 59.64             |
|                  | AD-SCAE | 97.92     | 61.03             |
|                  | HAT-SCAE| 95.85     | 60.98             |

among the three defense models. For clean samples from the Fashion-MNIST dataset, when using either the prior k-means classifier or the posterior k-means classifier, the classification accuracy of the HAT-SCAE is close to those of the AD-SCAE and the AT-SCAE, and it does not drop by more than 8.5% in comparison with that of the original SCAE. Therefore, the HAT has a small impact on the classification performance of the SCAE on clean samples.

The comparison of classification accuracy of the original SCAE and the three defense models (i.e., the AT-SCAE, the AD-SCAE and the HAT-SCAE) on adversarial samples is shown in Table 6. The bold numbers in the table also represent the best performance.

It is shown in the table that for adversarial samples from the MNIST dataset and the Fashion-MNIST dataset, when using either the prior k-means classifier or the posterior k-means classifier, the performance of the original SCAE decreases the most, which means that the original SCAE is vulnerable to the evasion attack proposed in this paper. The HAT-SCAE with either the prior k-means classifier or the posterior k-means classifier achieves the best classification accuracy (more than 81%) on the adversarial samples from the MNIST dataset. For adversarial samples from the Fashion-MNIST dataset, although the SCAE is not capable of fitting the dataset well and the effect of the defense models is limited, the HAT-SCAE with the prior k-means classifier also obtains the best performance, and the classification accuracy of the HAT-SCAE with the posterior k-means classifier is close to the best accuracy of the AT-SCAE. Therefore, the SCAE with the HAT can effectively defend against the evasion attack.

To visualize the classification performance of the original SCAE model, the baseline models (i.e., the AT-SCAE and the AD-SCAE) and our defense model HAT-SCAE on both clean samples and adversarial samples, the change in the classification accuracy of the models with the threshold of the perturbation amount on the MNIST dataset and the Fashion-MNIST dataset is shown in Figs. 2 and 3 respectively. The x-axis represents the threshold of the perturbation amount, and the y-axis represents the classification accuracy of the models. The intersections of the curves and the y-axis, where the value of the x coordinate is 0, which indicates that the input images are clean samples because there are no perturbations added to them, represents the classification accuracy of the models on clean samples. As the x coordinate value becomes larger, the clean samples turn into adversarial samples because there are adversarial perturbations added to the input images. The coordinates (x, y) on the curves represent that the classification accuracy of the models on the adversarial samples equals y when the perturbation added to each adversarial sample is not greater than the corresponding x,
Towards robust stacked capsule autoencoder with hybrid adversarial training

The higher the classification accuracy is when the perturbation amount threshold increases, the more robust the model.

It is shown in Fig. 2 that both the baseline models (i.e., the AT-SCAE and the AD-SCAE) and our defense model HAT-SCAE can improve the robustness of the SCAE against evasion attacks on the MNIST dataset, while the classification performance of the SCAE drops rapidly as the perturbation amount increases, which means the original SCAE model is vulnerable to the evasion attack. The AT-SCAE trains the model on adversarial samples and has a negative effect on the model accuracy on clean samples. Due to the advantage of distillation, the AD-SCAE significantly improves the model accuracy on clean samples, but at the cost that the vulnerabilities in the teacher model are transferred to the student model and weaken its robustness to a certain degree. The HAT-SCAE has better classification accuracy on both clean and adversarial samples, which proves HAT to be the best one among the three defense methods.

In Fig. 3 we can see that the difference in classification accuracy of the baseline models (i.e., the AT-SCAE and the AD-SCAE) and our defense model HAT-SCAE on the Fashion-MNIST dataset is not as obvious as that on the MNIST dataset. This is caused by the fitting ability of the SCAE. As adversarial training will decrease the accuracy of the model, we introduce the theory of distillation to optimize the training procedure, and the effect of distillation increases with the performance of the teacher model. Table 3 shows that the classification accuracy of the original SCAE model on the Fashion-MNIST dataset is worse than that on the MNIST dataset, as is the effect of the teacher model.

![Fig. 2](image1) The change in classification accuracy of the four models with the increase in the amount of perturbation on the MNIST dataset

![Fig. 3](image2) The change in classification accuracy of the four models with the increase in the amount of perturbation on the Fashion-MNIST dataset
Table 7 The comparison of classification accuracy of the original SCAE, the NTAT-SCAE and the HAT-SCAE on clean samples

| Classifier          | Model    | MNIST (%) | Fashion-MNIST (%) |
|---------------------|----------|-----------|-------------------|
| Prior k-means classifier | SCAE     | 97.40     | 67.71             |
|                     | NTAT-SCAE | 96.74     | 60.11             |
|                     | HAT-SCAE  | 95.22     | 59.17             |
| Posterior k-means classifier | SCAE     | 97.63     | 66.83             |
|                     | NTAT-SCAE | 96.94     | 61.74             |
|                     | HAT-SCAE  | 95.85     | 60.98             |

Under this circumstance, the three defense models (i.e., the AT-SCAE, the AD-SCAE and the HAT-SCAE) have similar classification accuracy on clean samples, but with the increase in the amount of perturbation, the performance of the AT-SCAE and the AD-SCAE decreases while the HAT-SCAE’s performance is relatively stable, which means that the HAT-SCAE is also the best among the three defense models.

6.4 Ablation study

Our proposed defense method HAT is the combination of adversarial training and adversarial distillation. In the training procedure of HAT-SCAE, the first step is adversarial distillation, and the second step is adversarial training. We provide an ablation study in this section, which replaces the first step with normal training to study the influence of adversarial distillation on the robustness of the HAT-SCAE to the evasion attack. We name the SCAE model obtained by the above method NTAT-SCAE (stacked capsule autoencoder with normal training and adversarial training).

We train the NTAT-SCAE model in the same experimental method and construct the prior k-means classifier and the posterior k-means classifier. The classification accuracy on clean samples is shown in Table 7. The bold numbers in the table represent the best performance.

From Table 7 we can obtain the following results: with either the prior k-means classifier or the posterior k-means classifier, the classification accuracies of the NTAT-SCAE and the HAT-SCAE on clean samples from both the MNIST dataset and the Fashion-MNIST dataset are close to each other, and the SCAE achieves the best performance on these datasets. As normal training does not add adversarial samples to the training dataset, the classification accuracy of the NTAT-SCAE is somewhat higher than that of the HAT-SCAE. The classification accuracy of the NTAT-SCAE and the HAT-SCAE is also close to that of the SCAE on the MNIST dataset and decreases by no more than 8.5% compared with the performance of the SCAE on the Fashion-MNIST dataset.

We test the robustness of the NTAT-SCAE to the evasion attack on adversarial samples from the MNIST and Fashion-MNIST datasets, and the results are shown in Table 8. The bold numbers in the table represent the best performance. In Table 8 it can be observed that the robustness of the NTAT-SCAE is worse than that of the HAT-SCAE, and the HAT-SCAE achieves the best performance.

To visualize the comparison between the NTAT-SCAE and the HAT-SCAE, we present diagrams to show the change in the classification accuracy with the perturbation amount threshold for the SCAE, the HAT-SCAE and the NTAT-SCAE, which are shown in Figs. 4 and 5. In the figures, the x-axis represents the threshold of the perturbation amount, and the y-axis represents the classification accuracy of the models. The intersections of the curves and the y-axis, where the value of the x coordinate is 0, which indicates that the input images are clean samples because there are no perturbations added to them, represent the classification accuracy of the models on clean samples. As the x coordinate value becomes larger, the clean samples turn into adversarial samples because there are adversarial perturbations added to the

Table 8 The comparison of classification accuracy of the original SCAE, the NTAT-SCAE and the HAT-SCAE on adversarial samples

| Classifier          | Model    | MNIST (%) | Fashion-MNIST (%) |
|---------------------|----------|-----------|-------------------|
| Prior k-means classifier | SCAE     | 0.38      | 3.82              |
|                     | NTAT-SCAE | 42.20     | 33.26             |
|                     | HAT-SCAE  | 81.36     | 37.38             |
| Posterior k-means classifier | SCAE     | 0.76      | 1.86              |
|                     | NTAT-SCAE | 51.90     | 33.92             |
|                     | HAT-SCAE  | 82.14     | 37.20             |
input images. Each coordinate \((x, y)\) on the curves represents that the classification accuracy of the model on the adversarial samples equals \(y\) when the amount of each perturbation added to the adversarial sample is not greater than the corresponding \(x\). The higher the classification accuracy is when the perturbation amount threshold increases, the more robust the model.

Figures 4 and 5 show the advantage of adversarial distillation on the robustness of the SCAE model. The experimental results of the HAT-SCAE on the MNIST dataset are obviously better than those of the NTAT-SCAE. On the Fashion-MNIST dataset, as the classification accuracy of the teacher model is imperfect, the performance of the HAT-SCAE is slightly better than that of the NTAT-SCAE.

From the above results, we can conclude that although the classification accuracy on clean samples is similar between the two defense models, the robustness of the HAT-SCAE is better than that of the NTAT-SCAE, which proves that adversarial distillation is necessary for the HAT to improve the robustness of the SCAE model.

7 Conclusion

In this paper, we propose an evasion attack method against the SCAE, which can cause the classifiers of the SCAE to output wrong predictions with a high probability. The evasion attack method confirms the existence of security vulnerabilities in
the SCAE with its high attack success rate and stealthiness. For this evasion attack, we further propose a defense method called HAT that combines adversarial training and adversarial distillation. HAT-SCAE, the robust model trained with HAT, not only ensures that the classification accuracy of the robust model is similar to that of the original SCAE model but also enhances the robustness of the SCAE against the evasion attack. We evaluate the performance of the HAT-SCAE with the experiments, and the experimental results show that among three defense models of the SCAE, HAT-SCAE, which has less impact on the accuracy of the model and achieves the best robustness, is the best model for enhancing the robustness of the SCAE. In our future work, we will further improve the robustness of the SCAE against black-box attacks and explore the application of the SCAE to security-sensitive tasks.

Appendix A: Generating adversarial samples for training

The generation algorithm of adversarial samples for training is based on the evasion attack algorithm proposed in Section 4. We replace the optimizer in the original algorithm with the following formula to accelerate the generation of adversarial samples:

\[
p'_{N+1} = p_N - \beta \cdot \text{sign} \left( \nabla_{p_N} \left( \|p_N\|_2 + \alpha \cdot f(x + p_N) \right) \right)
\]

(A1)

where \(p_N\) represents the perturbation obtained in the \(N^{th}\) iteration, \(f(x + p_N)\) is the target function of Formula (3), and \(\beta > 0\) is a hyperparameter that defines the step of update in each iteration. The whole algorithm is shown in Algorithm 5.

The update strategy of \(\alpha\) is the same as that described in Section 4. We will not go into much detail here.

Acknowledgements The research of this paper was supported by Natural Science Foundation of Shanghai Municipality (Grant NO.22ZR142-2600).

Declarations

Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Kosiorek A, Sabour S, Teh YW, Hinton GE (2019) Stacked capsule autoencoders. Adv Neural Inf Process Sys 32
2. Yoon J (2017) Adversarial Attack to Capsule Networks. Published: [EB/OL]
3. Michels F, Uelwer T, Upshulte E, Harmeling S (2019) On the vulnerability of capsule networks to adversarial attacks. arXiv preprint. arXiv:1906.03612
4. Marchisio A, Nanfa G, Khalid F, Hanif MA, Martina M, Shafique M (2019) CapsAttacks: robust and imperceptible adversarial attacks on capsule networks. arXiv preprint. arXiv:1901.09878
5. De Marco A (2020) Capsule networks robustness against adversarial attacks and affine transformations. PhD Thesis, Politecnico di Torino Turin, Italy
6. Dai J, Xiong S (2020) An evasion attack against stacked capsule autoencoder. arXiv preprint. arXiv:2010.07230
7. Li Y, Su H, Zhu J (2021) AdvCapsNet: to defense adversarial attacks based on Capsule networks. J Vis Commun Image Represent 75:103037. Publisher: Elsevier
8. Peer D, Stabinger S, Rodriguez-Sanchez A (2019) Increasing the adversarial robustness and explainability of capsule networks with \(\gamma\)-capsules. arXiv preprint. arXiv:1812.09707
9. Garg S, Alexander J, Kothari T (2017) Using thermometer encoding to defend against adversarial attacks. Proceedings of the CS229 final project session, Stanford, CA
10. Sabour S, Frost N, Hinton GE (2017) Dynamic routing between capsules. Adv Neural Inf Process Sys 30
11. Hinton GE, Sabour S, Frost N (2018) Matrix capsules with EM routing. In International conference on learning representations
12. Lee J, Lee Y, Kim J, Kosiorek A, Choi S, Teh YW (2019) Set transformer: a framework for attention-based permutation-invariant
55. Fan W, Sun G, Su Y, Liu Z, Lu X (2019) Hybrid defense for deep neural networks: an integration of detecting and cleaning adversarial perturbations. In: 2019 IEEE international conference on multimedia & expo workshops (ICMEW), pp 210–215. IEEE
56. Ruan Y, Dai J (2018) TwinNet: a double sub-network framework for detecting universal adversarial perturbations. Futur Internet 10(3):26. Publisher: Multidisciplinary Digital Publishing Institute
57. AprilPyone MM, Kinoshita Y, Kiya H (2019) Filtering adversarial noise with double quantization. In: 2019 Asia-pacific signal and information processing association annual summit and conference (APSIPA ASC), pp 1745–1749. IEEE
58. Kuhn HW (1955) The Hungarian method for the assignment problem. Nav Res Logist Q 2(1-2):83–97. Publisher: Wiley Online Library
59. Hinton GE, Vinyals O, Dean J, others (2015) Distilling the knowledge in a neural network 2(7). arXiv preprint. arXiv:1503.02531
60. Kosiorek A, Sabour S, Teh YW, Hinton GE (2019) Stacked capsule autoencoders. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No.: 1390, pp 15512–15522

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Jiazhu Dai received the M.S. and Ph.D. degrees in Computer Science and Technology from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He was a Visiting Scholar at Data Science Institute, Imperial College London, United Kingdom, supported by the China Scholarship Council, from 2017 to 2018. He is currently an Associate Professor at the School of Computer Engineering and Science, Shanghai University, China. He has been involved in extensive research works in the fields of information security. His current research interests include mobile and IoT security, security and privacy in machine learning, applications of deep learning in systems security, and applied cryptography.

Siwei Xiong received the bachelor’s degree in Computer Science from Shanghai University, China, in 2019; he is currently a graduate student at the School of Computer Engineering and Science, Shanghai University, China.