Review of Adversarial Attack and Defense Techniques in Computer Vision

Zibai Wang
Department of Computer Science
University of Waterloo
Waterloo, ON
zibai.wang@uwaterloo.ca

Abstract

When computer vision is intertwined with machine learning, it can produce numerous applications, such as autonomous cars and video surveillance. Recent studies show security concerns since these applications are vulnerable to adversarial attacks. This paper discusses various adversarial attacks and defensive techniques against them. Finally, the advantages and disadvantages of different techniques will be examined and an evaluation will be provided.

1 Introduction

1.1 Computer vision

The application domain: computer vision is the application of artificial intelligence techniques to extract information from images. The purpose of this domain normally is to resolve a problem or understand a scenario. There are various industries that can utilize computer vision systems, such as autonomous vehicles and image processing (Danuser, 2011). The result that a computer vision system generates is an abstract illustration of the input image. From this illustration, the image processing system can start to extract information. This new application domain is also sometimes vulnerable to adversarial attacks, thus raise some security concerns. For example, there is a security threat when an autonomous car recognizes a stop sign into something else due to an adversarial attack.

1.2 Adversarial attack

Adversarial attacks are attacks by feeding Adversarial Examples into Neural Network to cause machine learning to make mistakes. For example, a small perturbation was added by attackers on an image of a panda and this makes the machine recognize it as a gibbon with high confidence (Goodfellow et al., 2014).

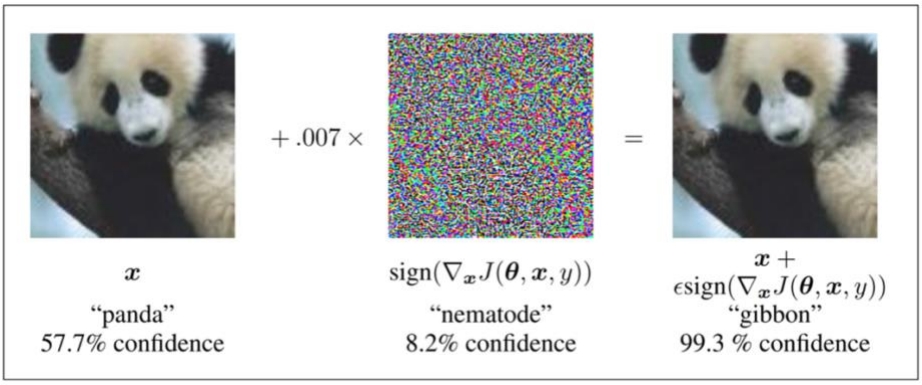

Figure 1: After applying on layer of unnoticeable perturbation,
"panda" is perceived as "gibbon". (Image Credit: (Goodfellow et al., 2014)

2 Attack methods

There are three types of image adversarial attacks, attacks in the training stage, testing stage and model deployment stage. This paper will discuss the attack in the training phase briefly and testing phase in details.

2.1 Training phase

The training model is often interrupted by the training phase attack. An ideal attack is to infect all the training data. There are three groups of attack approaches.

The first one is data infusion. When the adversary could not retrieve training data and algorithms, he could distort the training model by adding adversarial examples.

The second group is data modification. When the adversary has no contact with training algorithms but can retrieve the training dataset. The adversary can modify the data to be used to train the model.

The third group is logic distortion. The adversary can intervene in the training algorithms. Barreno et al. (2006) claimed this corruption attacks first. Their research experiment deleted the raining dataset to make the logic of the algorithm change accordingly.

2.2 Testing phase

2.2.1 White-box scenario

In white-box attacks, the adversaries have access to the structure and parameters of the target model. An adversary can analyze the structure of the target system and build adversarial samples accordingly to carry out attacks. The work (Papernot et al., 2016) proposed an adversarial crafting framework as shown in the below picture.

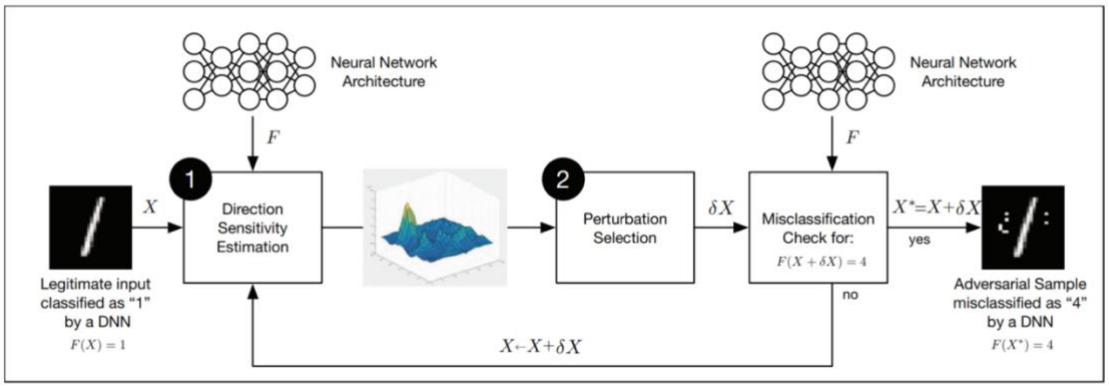

Figure 2: (Image Credit: Papernot et al., 2016)

Goodfellow et al.(2014) created a method called "fast gradient sign method" to generate adversarial examples for the attack. This method is one of the first and most popular adversarial attack methods.

The formulation is:

Let $\boldsymbol{\theta}$ be the parameters of a model, $\boldsymbol{x}$ the input to the model, y the targets associated with $\boldsymbol{x}$ (for machine learning tasks that have targets) and $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ be the cost used to train the neural network. We can linearize the cost function around the current value of $\boldsymbol{\theta}$, obtaining an optimal max-norm constrained pertubation of

$$\boldsymbol{\eta} = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

Figure 2: (Image Credit: (Goodfellow et al., 2014))

The fast gradient sign method creates adversarial examples by calculating the gradients of the neural network. Given an input image, this method creates a new adversarial image that maximizes the loss by using the gradients of the loss with respect to the input image.

It's interesting that the gradients are taken with respect to the input image since our purpose is to create an image that has the maximum loss. One way to achieve this is to check how much each pixel in the image contributes to the loss value and add a perturbation accordingly. Chain Rule can make this process extremely fast. (TensorFlow)

Code example

```
def create_adversarial_pattern(input_image, input_label):
  with tf.GradientTape() as tape:
    tape.watch(input_image)
    prediction = pretrained_model(input_image)
    loss = loss_object(input_label, prediction)

  # Get the gradients of the loss w.r.t to the input image.
  gradient = tape.gradient(loss, input_image)
  # Get the sign of the gradients to create the perturbation
  signed_grad = tf.sign(gradient)
  return signed_grad
```

Figure 3: (Image Credit: (TensorFlow))

It changed the original input from Labrador retriever to bath towel:

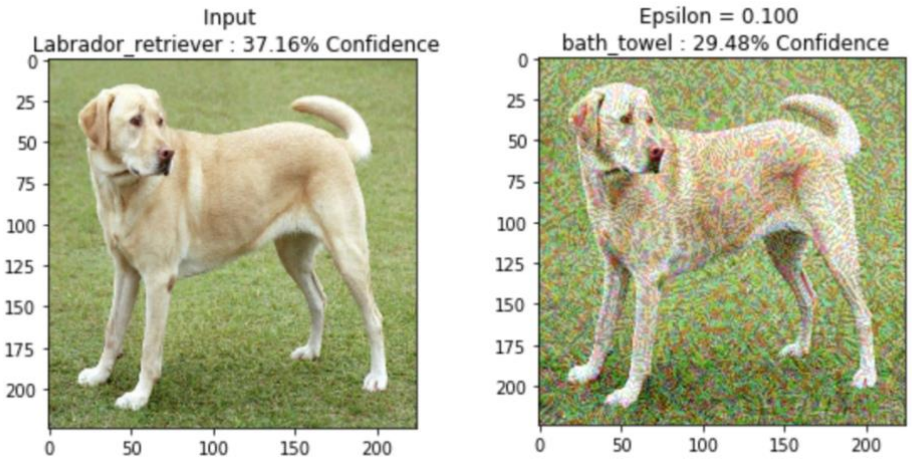

Figure 4: (Image Credit: (TensorFlow))

2.2.2 Black-box scenario

In black-box attacks, adversaries have no knowledge about the target model. They evaluate the input/output pairs to analyze the model's vulnerability. There are three different types of black-box scenarios: Utilizing Transferability, Model Inversion, Model Extraction (Qiu et al., 2019).

The substitute model is one of the black-box attack techniques. It can be used when the adversary has no access to the internal logic of the model. This substitute model can be trained to imitate the logic of the target model through APT interfaces (Papernot et al., 2016). When the substitute model has good performance, an adversary can use white-box attack techniques on the substitute model to

produce adversarial examples to mislead the target model.

The work of Papernot et al.(2016) was the first to introduce an effective algorithm to attack DNN classifiers, under the condition that the adversary has no access to the classifier's parameters or training set (black-box). The research of Papernot and his coworkers was also the first to propose a new effective algorithm to attack DNN classifiers in the black-box scenarios. An adversary can only obtain the output label by proving the input to the classifier. Only partial knowledge about the classifier is provided to the adversary, for example, the input data type, like "photographs, digits, etc..", the basic architecture of the classifier (e.g., CNNs, RNNs).

As Qiu et al. (2019) mentioned, "Their work used cross-model transferability of adversarial samples to carry out black-box attack, which used synthetic input generated by the adversary to train a local substitute model and used this substitute model to make adversarial samples, which could be misclassified by the original target model." (11).

Specifically, there are four main steps (Han et al., 2019).

1. Replicate the training set. For instance, in order to attack a victim system that is designed to classify hand-written digits, it needs to make an initial substitute training set by getting samples from the test and/or handcrafted samples.

2. Feed the replica training set X into the victim classifier and obtain their output label Y. Choose one of the substitute DNN models to train on (X, Y) to get F'. The chosen DNN must have similar structures to the victim system.

3. Increase the dataset (X, Y) and iteratively retrain the substitute model F'. This will increase the accuracy of substitute model F'.

4. Utilize white-box attack methods, such as FGSM that we discussed before to attack model F'. The adversarial examples generated by this attack are most likely to cause the target model F to output the wrong result, by the property of "transferability."

3 Defense techniques

3.1 Adversarial training

Remember that we discussed using FGSM to generate attack samples in the previous section, (Goodfellow et al., 2014) also mentioned that we can use FGSM to generate adversarial examples and feed these examples into training process with the correct label. This is called Adversarial training. It can teach the machine to identify these adversarial examples and classify future examples with a higher success rate. (Kurakin et al., 2016) improved the training strategy to make it scalable to a larger dataset such as Image Net.

3.2 Gradient hiding

This technique conceals model gradient information, so that the adversary will not know. For example, a gradient-based attack is worthless in front of non-differentiable models, such as a decision tree (Tramèr et al., 2017).

3.3 Blocking the transferability to prevent the black-box attack

As we discussed in the black-box attack, many black-box attack methods assume the target system has the property of transferability.. Hosseini et al. (2017) introduced a training method that can block the transferability. "As the input is more perturbed, the classifier smoothly outputs lower confidence on the original label and instead predicts that the input is "invalid". They augment the output class set with a NULL label and train the classifier to reject the adversarial examples by classifying them as NULL." (Hosseini et al., 2017) With the help of NULL labelling, the system learned to distinguish between the clean data and adversarial data, and instead of classifying the adversarial data to the correct label, it discarded the adversarial examples. The experiment showed that this way effectively blocks the transferability of the system.

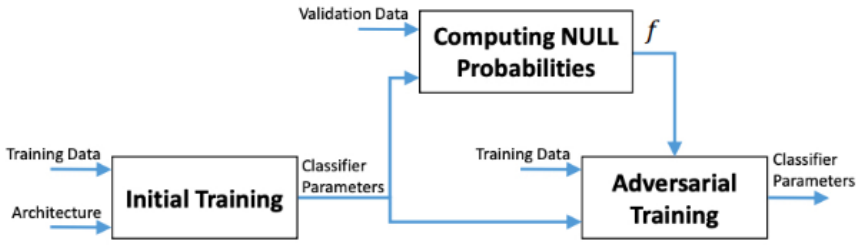

Figure 5: (Image Credit: (Hosseini et al., 2017))

3.4 Data compression

JPG compression method can be used to defense FGSM attack by enhancing network model recognition accuracy (Das et al., 2017). However, this method cannot be used to hold against intense attacks. The constraint is due to the method itself. A huge number of compression would cause the precision of original image classification to reduce, whereas a smaller degree of compression is not sufficient to eliminate the disturbance.

3.5 Feature squeezing

This model improvement technique is utilized by decreasing the density of the data sample, thus diminish the adversarial impact. There are two experimental ways, the first one is to decrease the pixel level color depth. For example, the experimenter can program the color with fewer values. The second way is to utilize a smooth filter on the images. For instance, there are numerous inputs added to a value that makes the model stronger under attack. The disadvantage of this technique is that by reducing the pixel value, the accuracy of sample recognition is also decreased.

3.6 Defense Generative Adversarial Nets (defense-GAN)

Samangouei et al. (2018) discussed that defense generative adversarial nets can decrease the efficiency of adversarial perturbance from both white-box and black-box attacks.

Defense-GAN suggests using a new framework that can leverage the expressive capability of the generative model. Its main algorithm is to “project” input images onto the range of the generator G by minimizing the reconstruction error before sending the image to the classifier. As a result, the clean data sample will be closer to the range of G, comparing to the adversarial samples, which can decrease the adversarial perturbance. The workflow of how Defense-GAN works is shown below.

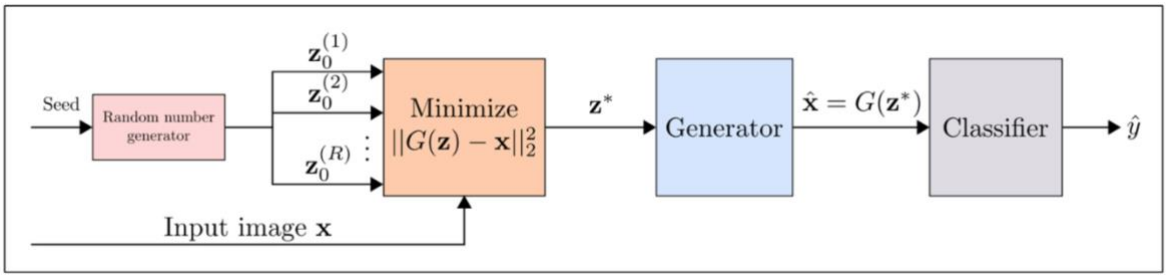

Figure 6: (Image Credit: (Samangouei et al., 2018))

3.7 High-Level Representation Guided Denoiser (HGD)

Liao et al. (2017) introduced HGD as a defense for image classification and this won first place defend

in NIPS competition 2018. Compared with standard denoiser, HGD overcomes the error amplification effect issue by applying a loss function representing the difference between the denoised image output and clean image output. The below picture shows the idea of HGD. Although there is only a tiny difference between the original image and adversarial image, it can be amplified in the high-level representation (for example, logits) of a CNN. HGD utilizes the distance over high-level representations to instruct the training process of an image denoiser to neutralize the effect of adversarial perturbation.

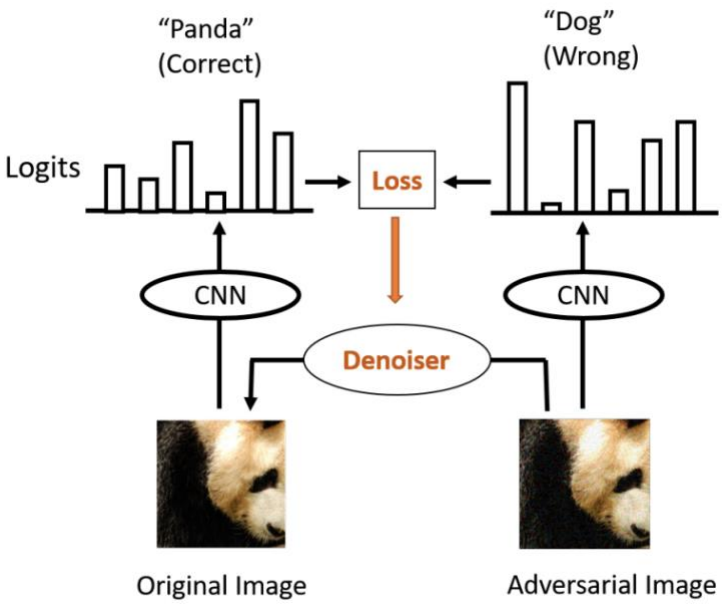

Figure 7: (Image Credit: (Liao et al., 2017))

3.8 Free adversarial training

Shafahi et al. (2019) present an algorithm that greatly reduces the overhead cost of generating adversarial examples by recycling the gradient information computed when updating model parameters. Here is a code sample representing this idea

Algorithm 1 "Free" Adversarial Training (Free-m)

Require: Training samples X, perturbation bound ϵ, learning rate τ, hop steps m

Initialize θ
$\delta \leftarrow 0$
for epoch $= 1 \ldots N_{ep}/m$ **do**
  for minibatch $B \subset X$ **do**
    for i $= 1 \ldots m$ **do**
      Update θ with stochastic gradient descent
        $g_\theta \leftarrow \mathbb{E}_{(x,y)\in B}[\nabla_\theta l(x+\delta, y, \theta)]$
        $g_{adv} \leftarrow \nabla_x l(x+\delta, y, \theta)]$
        $\theta \leftarrow \theta - \tau g_\theta$
      Use gradients calculated for the minimization step to update δ
        $\delta \leftarrow \delta + \epsilon \cdot \text{sign}(g_{adv})$
        $\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$
    end for
  end for
end for

Figure 8: (Image Credit: (Shafahi et al.2019))

4 Analysis of defense techniques

Since Adversarial Attack remains an actively growing topic, new attack and defense techniques are

being developed every day. This paper is designed to give a brief introduction of some interesting methods only (selected by the author).

Adversarial Training is one of the few defenses against adversarial attacks that withstands strong attacks (Shafahi et al. 2019), the trained classifier has good robustness on FGSM attacks but it is still vulnerable to iterative attacks (Han et al. 2019). My paper has not covered iterative attacks, one example of iterative attack is Projected Gradient Method (PGD) attack (Madry et al. 2017). A variant of Adversarial training has been developed to address this issue, it is called Adversarial Training with PGD (Madry et al. 2017). However, with extra training of PGD, this solution is hard to scale to a large data set as ImageNet. Therefore, another variant of it, Ensemble Adversarial Training can protect CNN models against single-step attacks and can be also applied to large datasets such as ImageNet Training (Tramer et al. 2017). Although Ensemble Adversarial Training is more efficient than the previous two versions, comparing with normal training (without adversarial samples), it still has a very high cost. As a motivation, the fourth variant, free adversarial training algorithm (discussed in the previous section) was created. From the experiment result of Shafahi and his colleagues' work, they compared the (free adversarial trained) model with the PGD trained model (variant 2), PGD only achieved a slightly better result (4.5%) but with 340% longer training time. (2019)

Now, I have compared some variants of Adversarial training defense methods. As one of the earliest, strongest, most invested methods, they often serve as a performance baseline when new techniques are being developed.

There is a common issue with Adversarial training, because of Adversarial examples are generated using one or more chosen attack models and added to the training set, adversarial training does not perform very good when a different attack strategy is used by the attacker. As a motivation, Defense-GAN was developed. This solution is designed for handling any classification model and any type of attack since it makes no assumption of how the adversarial examples were generated. It performs very well in the evaluation; however, it has its own disadvantages. (Samangouei et al., 2018) mentioned that "The success of Defense-GAN relies on the expressiveness and generative power of the GAN. However, training GANs is still a challenging task and an active area of research, and if the GAN is not properly trained and tuned, the performance of Defense-GAN will suffer on both original and adversarial examples."

Lastly, I would like to discuss the performance of HGD (High-Level Representation Guided Denoiser), the winner of defense in NIPS competition 2018. As mentioned in (Liao et al., 2017) work, compared to Adversarial training, HGD uses much less training images and run much faster than Adversarial training, however it still outperforms Adversarial training defending various attacks.

5 Conclusion

Attacks during the training stage are not common in the real world since it's relatively easy to defend it by restricting access to the training data. For the attacks in the testing stage, compared to white-box attacks, black box attacks have better applicability in the real world since it requires no information about the target model. (Qiu et al., 2019)

To the best of my knowledge, there is no existing defense method that is provable to be able to fully defend all types of the Adversarial attacks. Most of the defense techniques are limited to defend some specific types of attacks and have scalability issues. Moreover, new attack techniques are being developed to conquer these defense technologies. Based on my survey, the best defense method I found is HGD (High-Level Representation Guided Denoiser), which has high scalability and can defend various attacks.

This review paper gives a brief introduction about adversarial attack and defense method in the computer vision field. I hope my work can attract more researchers into this field and help make progress.

References

[1] “Adversarial example using FGSM.” TensorFlow, https://www.tensorflow.org/tutorials/generative/adversarial_fgsm

[2] Barreno, M.; Nelson, B.; Sears, R.; Joseph, A.D.; Tygar, J.D. Can machine learning be secure? In Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, Taipei, Taiwan, 21–24 March 2006:16–25.

[3] Danuser, Gaudenz. "Computer Vision in Cell Biology." *Cell* 147.5 (2011): 973-978.

[4] Das, N.; Shanbhogue, M.; Chen, S.T.; Hohman, F.; Chen, L.; Kounavis, M.E.; Chau, D.H. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. In: arXiv preprint arXiv:1705.02900 (2017).

[5] Geng, Daniel and Veerapaneni Rishi. “Tricking Neural Networks: Create your own Adversarial Examples”. Medium. 17 March, 2019, https://medium.com/@ml.at.berkeley/tricking-neural-networks-create-your-own-adversarial-examples-a61eb7620fd8

[6] Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples, 2014

[7] Goodfellow, Ian., et al. “Attacking Machine Learning with Adversarial Examples”. OpenAI, 24 February, 2017, https://openai.com/blog/adversarial-example-research/

[8] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil Jain. Adversarial attacks and defenses in images, graphs and text: A review. 2019.

[9] Hosseini, H.; Chen, Y.; Kannan, S.; Zhang, B.; Poovendran, R. Blocking transferability of adversarial examples in black-box learning systems. 2017

[10] Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[11] Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Zhu, J.; Hu, X. Defense against adversarial attacks using high-level representation guided denoiser. 2017.

[12] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. 2017.

[13] Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016:582–597.

[14] Qiu, S., Liu, Q., Zhou, S., and Wu, C. “Review of artificial intelligence adversarial attack and defense technologies,” *Applied Sciences.* 9 (2019): 909-938.

[15] Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. *arXiv* 2018, arXiv:1805.06605.

[16] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! arXiv preprint arXiv:1904.12843, 2019.

[17] Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. arXiv 2017.

[18] Zantedeschi, V., et al. “Efficient Defenses Against Adversarial Attacks”. arXiv preprint arXiv:1707.06728 (2017)

[19] Zawistowski, Pawel. “Adversarial Examples: A Survey.” Baltic URSI Symposium (2018)