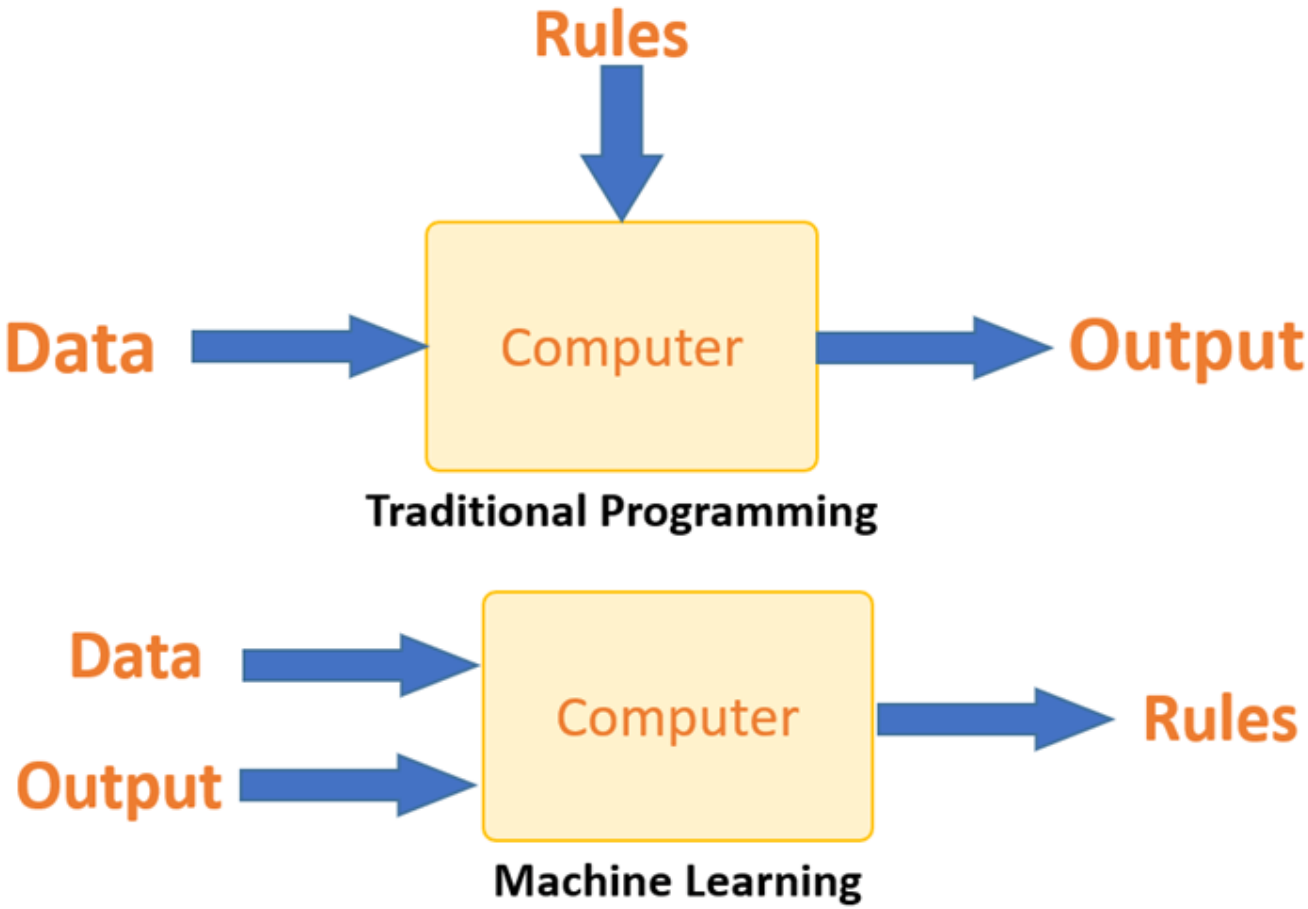
Makine Öğrenmesi Modellerinde Adversarial Saldırılar

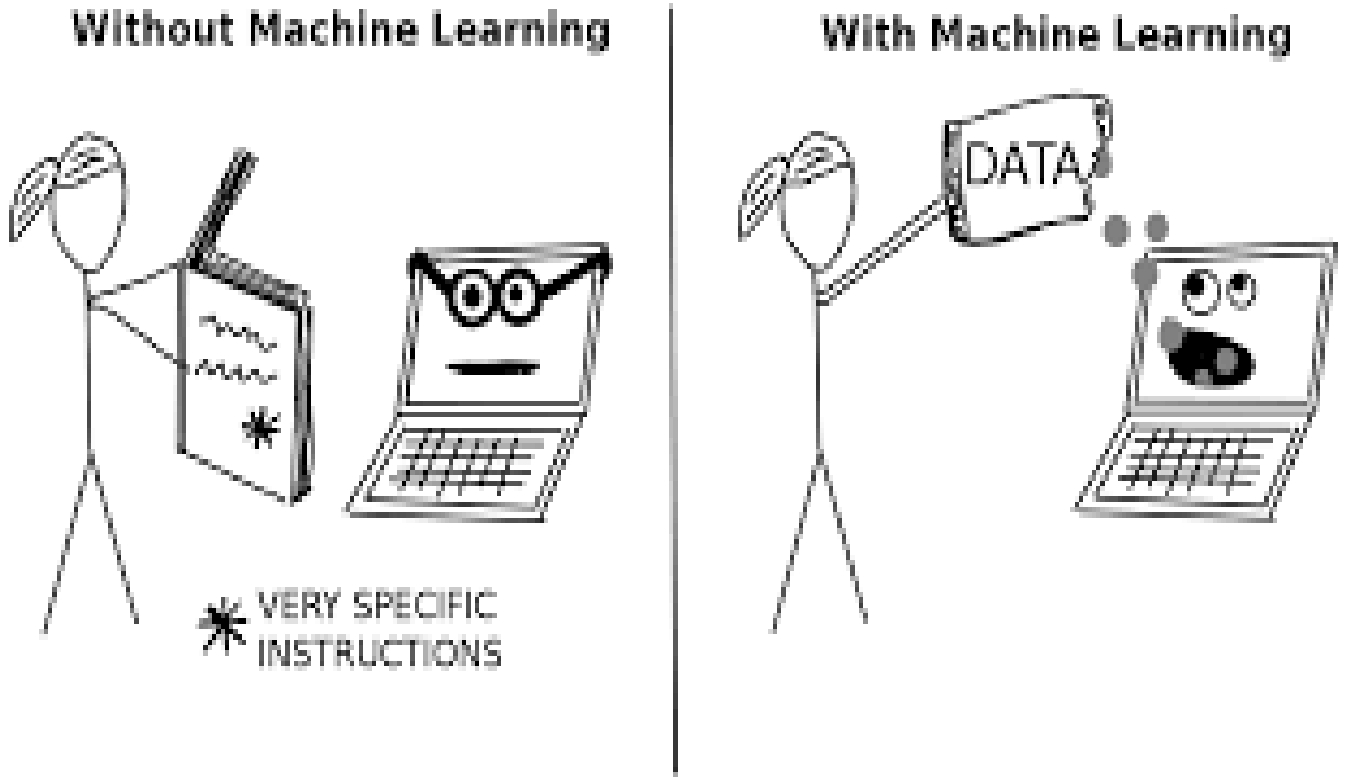
–

Adversarial Machine Learning

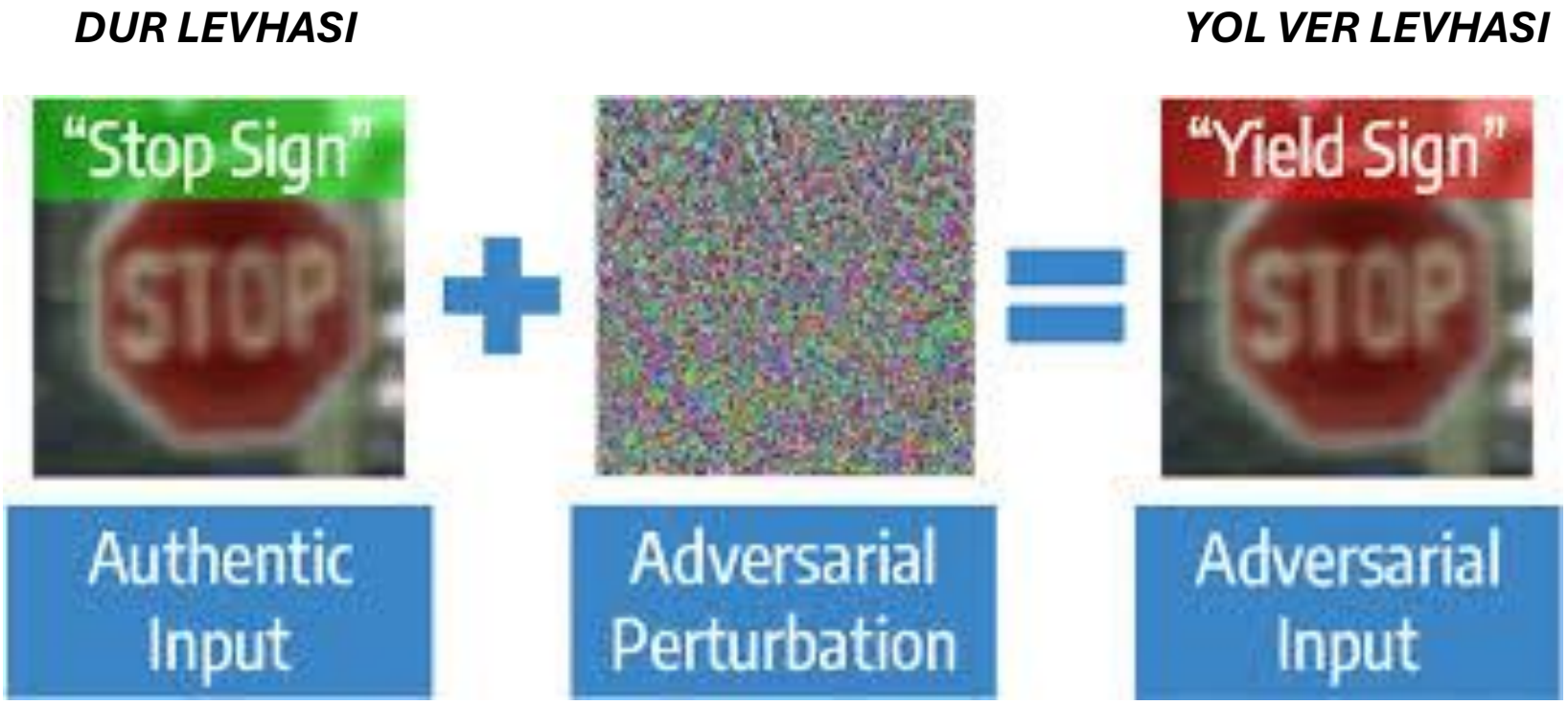
Makine Öğrenmesi Nedir?



Karmaşık talimatlar yerine; Makine fazlaca veriden talimatları kendisi çıkarıyor

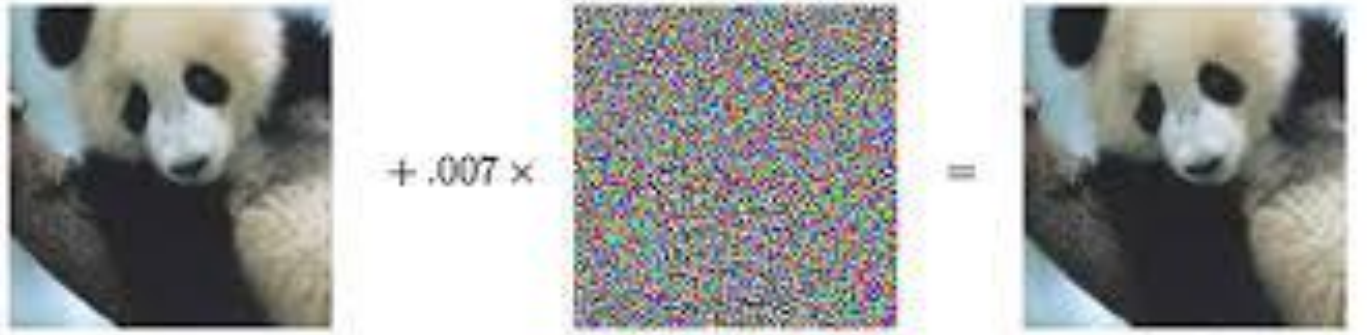


Kendi keşfettiği desenleri anlamlandırarak öğrenme işlemini gerçekleştiriyor, dolayısıyla eklenecek bir gürültü insan gözüyle yanılsamaya sebep olmasa da makine için şaşırtıcı derecede farklı sonuçlar bulunmasına sebep olabilir (bir adversarial saldırı örneği).

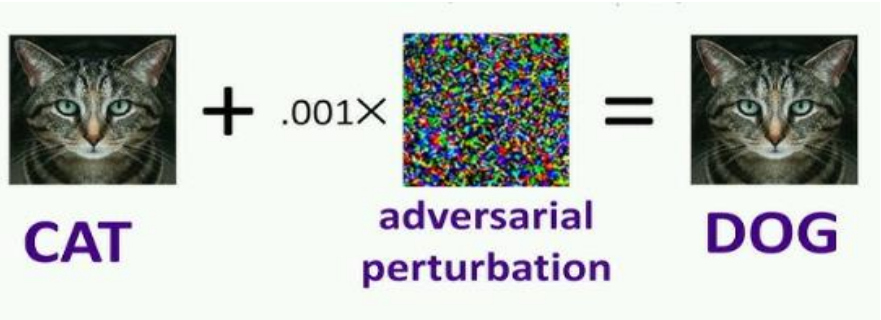


Adversarial Machine Learning

- ML modellerini *manipüle* etmeye çalışan bir disiplin.



Types of Adversarial attacks



1. Saldırganın Bilgi Düzeyine Göre (Knowledge-based)

*** Beyaz Kutu (White-box):** Saldırgan modelin her şeyini bilir; mimarisi, algoritması, ağırlıkları (weights) ve gradyanları. *Uygulama yapacağımız FGSM buna bir örnektir.*

- **Siyah Kutu (Black-box):** Saldırgan modelin iç yapısını bilmez. Sadece modele bir girdi (input) verir ve çıkan sonucu (output) görür. *Dipnot: Siyah kutu saldırıları genellikle "Transferability" (Aktarılabilirlik) özelliğini kullanır. Saldırgan kendi evinde benzer bir model eğitir, ona saldırır, sonra aynı saldırıyı hedef modele uygular.*

*** Gri Kutu (Grey-box):** Saldırgan model hakkında kısmi bilgiye sahiptir (örneğin sadece mimariyi biliyor ama ağırlıkları bilmiyordur).

2. Saldırının Zamanlamasına Göre (Timing-based)

- **Evasion (Kaçınma) Saldırıları:** Model eğitilmiştir ve çalışıyordur. Saldırgan, modeli o anki bir karar için yanıltmaya çalışır.

Örnek: Bir spam filtresini geçmek için e-postaya görünmez karakterler eklemek.

- **Poisoning (Zehirleme) Saldırıları:** Model henüz eğitim aşamasındayken yapılır. Saldırgan, eğitim veri setine "kirli" veriler sızdırır.

Örnek: Modelin belirli bir etiketi (örneğin bir logoyu) her gördüğünde yanlış karar vermesini sağlayacak bir "arka kapı" (backdoor) yerleştirmek.

3. Saldırganın Amacına Göre (Goal-based)

- **Hedefli (Targeted) Saldırı:** Saldırgan, modelin spesifik bir çıktı vermesini ister.
- Örnek: "Bu dur tabelasını dur tabelası olarak görme, hız sınırı olarak gör."
- **Hedefsiz (Untargeted/Non-targeted) Saldırı:** Saldırganın tek amacı modelin yanlış yapmasıdır, sonucun ne olduğu önemli değildir**.**
- Örnek: "Bu dur tabelasını ne olarak görürsen gör ama dur tabelası olarak görme."

Korunma Yöntemleri

- **Adversarial Training (en etkililerden)**

Model, adversarial örneklerle yeniden eğitilir. Bir «aşı» mantığıdır. Örneğin FGSM gibi yöntemlerle üretilmiş manipüle etmeye yönelik örnekleri de gösteririz. Böylece model bağışıklık kazanır.

- **Savunma Amaçlı Damıtma**

Model kararları daha esnek hale getirilir. Çok keskin (%99 köpek, %1 kedi) gibi keskin gradyanlar saldırganlar için kolaylık sağlar. Distillation ile model çıktıları daha yumuşak (%70 kedi, %20 köpek, %10 kuş) hale getirilir. Sonuç olarak modelin küçük girdi değişikliklerine (gürültüye) hassasiyeti azalır.

- **Girdi Dönüştürme, ön işleme**

Resim modele girmeden önce üzerindeki olası gürültüler temizlenmeye çalışılır.
Yöntemler: Bit Depth Reduction, JPEG Sıkıştırma, Blurring

- **Adversarial Detection**

Saldırıyı engellemek yerine girdinin bir saldırı olup olmadığı sınıflandırılmaya çalışır.

Nasıl? Ayrı bir bekçi gibi model eğitilir bu modelin tek görevi gelen resmin doğal mı yoksa oynanmış bir örnek mi olduğunu ayırt etmektir.

Dünyadan Adversarial ML örnekleri

- 🛑 **1. Otonom Araçlar & Trafik Levhaları**

STOP tabelasına küçük çıkartlamalar ekleyerek, görüntüyü insanlar için hala STOP yaparken modelin SPEED LIMIT 45 olarak algılaması sağlandı.

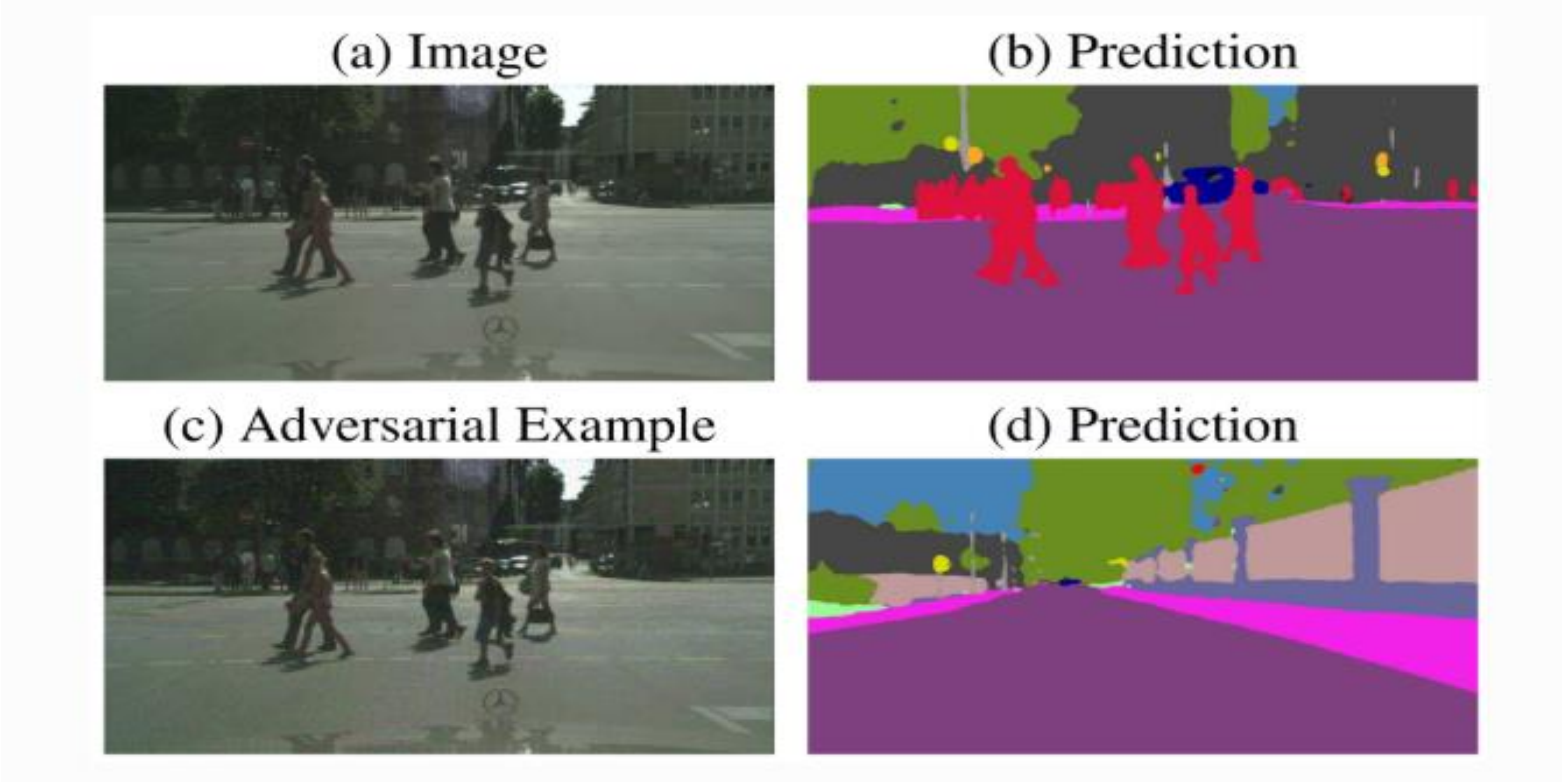
b) Attack on the **traffic sign classification**: a stop sign misclassified as a speed limit 45km/h sign by LISA-CNN [5].

https://www.catalyzex.com/paper/adversarial-attacks-on-traffic-sign

2. Otonom Sürüş ve Güvenlik Riski (Yayaları görmezden gelme)

Otonom bir aracın kamera sistemine sızan bir saldırgan, görüntüye çok küçük gürültüler ekleyebilir.

Normalde sistem tarafından tespit edilen yayalar bu gürültü sonrası yayaları tespit edemez hale geliyor



3. OCR & Belge Okuma Sistemleri

Fatura ve belgelerde harflere çok küçük gürültüler eklendi, insan gözüyle yine aynı okunabilen belgeler OCR modeli tarafından tamamen yanlış okundu. (Evasion Attack)

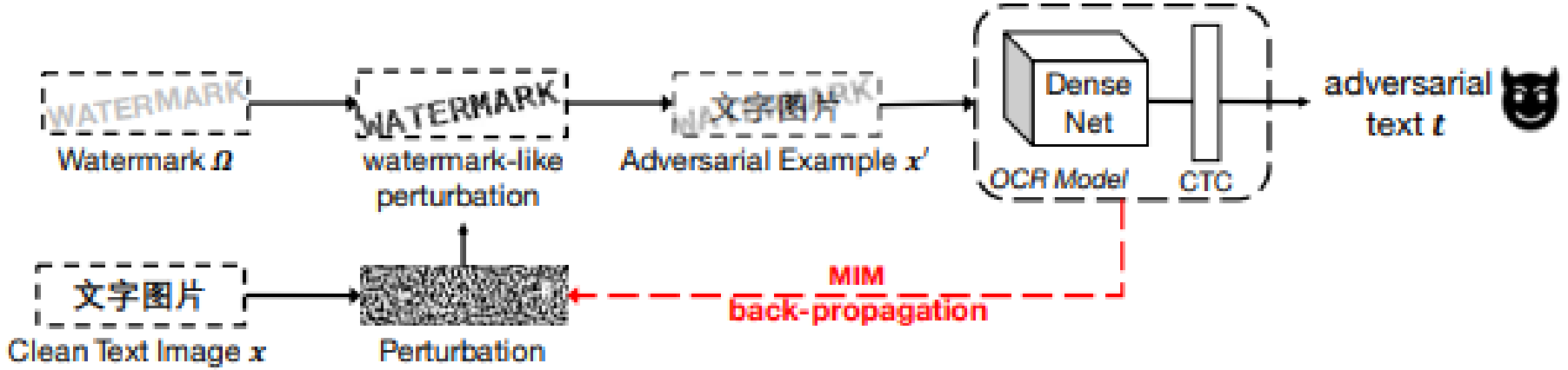


Figure 1: The pipeline of the WATERMARK attack. We generate noise using MIM with CTC loss function back propagating the targeted DenseNet and then mask the noise outside the watermark region. We iterating the procedure above until an iterations threshold.

*"WATERMARK (Filigran) saldırısının iş akışı (pipeline) şöyledir: Hedeflenen **DenseNet** modeli üzerinden **CTC kayıp fonksiyonu** ile geri yayılım (backpropagation) yaparak **MIM** (Momentum Iterative Method) aracılığıyla gürültü üretiyoruz ve ardından bu gürültüyü filigran bölgesi dışındaki alanlarda maskeliyoruz. Belirli bir yineleme eşiğine (iterations threshold) ulaşana kadar yukarıdaki prosedürü tekrarlıyoruz."*

https://arxiv.org/abs/2002.03095

	original	MIM	WM	WM_{init}	WM_{neg}	WM_{edge}	OCR output	English translation
substitution	靠左行驶	靠左行驶	靠左行驶	靠左行驶	靠左行驶	靠左行驶	靠右行驶	drive left→ drive right
	9月1号	9月1号	9月1号	9月1号	9月1号	9月1号	9月9号	1 Sep.→9 Sep.
	我叫小方	我叫小方	我叫小方	我叫小方	我叫小方	我叫小方	我叫孙方	I am Xiao Fang→ I am Sun Fang
	hire	hire	hire	hire	hire	hire	fire	hire→fire
	一点下课	一点下课	一点下课	一点下课	一点下课	一点下课	二点下课	class is over at 1→ class is over at 2
−	一点下课	一点下课	一点下课	一点下课	一点下课	一点下课	一下课	class is over at 1→ once class is over
+	一点下课	一点下课	一点下课	一点下课	一点下课	一点下课	一点不下课	class is over at 1→ class is not over at 1

Table 1: Adversarial examples with different attacks. The last two rows show text deletion and insertion. Other rows show text substitution.

Uygulama: **FGSM (Fast Gradient Sign Method):**

- *Kullanılan Veri Seti:* **MNIST** (28×28 gri seviye el yazısı rakamlar), 60.000 train, 10.000 test
- *Model Mimarisi* : Feedforward NN, 1 Gizli Katman, 128 nöron, ReLU, output: 10 sınıf Softmax
- *Saldırı Türü:* **FGSM (Fast Gradient Sign Method):** White-box, Evasion, Untargeted saldırı.

$$x^{adv} = x + \varepsilon.sign(\nabla_x J(x, y_{true}))$$

$$\begin{aligned} &where \\ &x \quad - \text{ Clean Input Image} \\ &x^{adv} \quad - \text{ Adversarial Image} \\ &J \quad - \text{ Loss Function} \\ &y_{true} \quad - \text{ Model Output for x} \\ &\varepsilon \quad - \text{ Tunable Parameter} \end{aligned}$$

Modelin hatasını en hızlı artıran yön, yani gradyanın işareti alınır. Girdi bu yönde çok küçük bir epsilon kadar kaydırılır. Bu küçük kayma modelin karar sınırını geçmesine yeter.

Figure 1

Normal prediction: 9

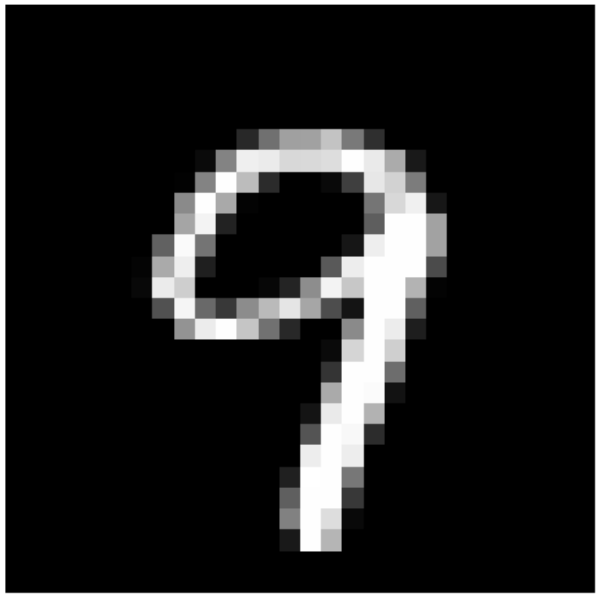
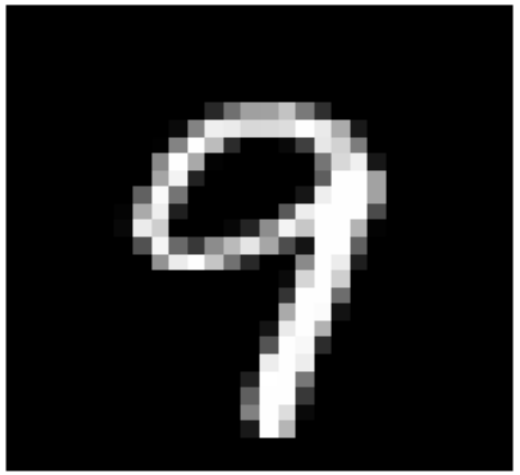


Figure 1

x=18.8 y=5.2
[0.000]

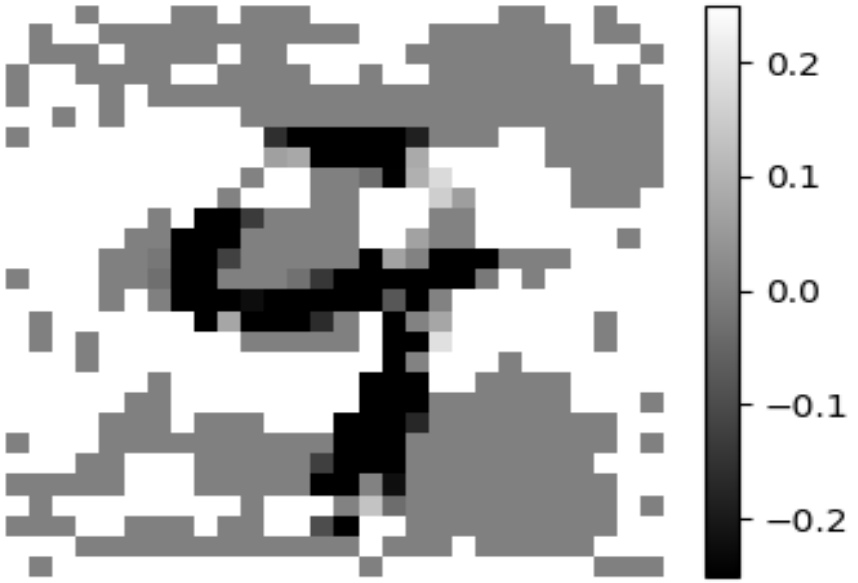
Original Image

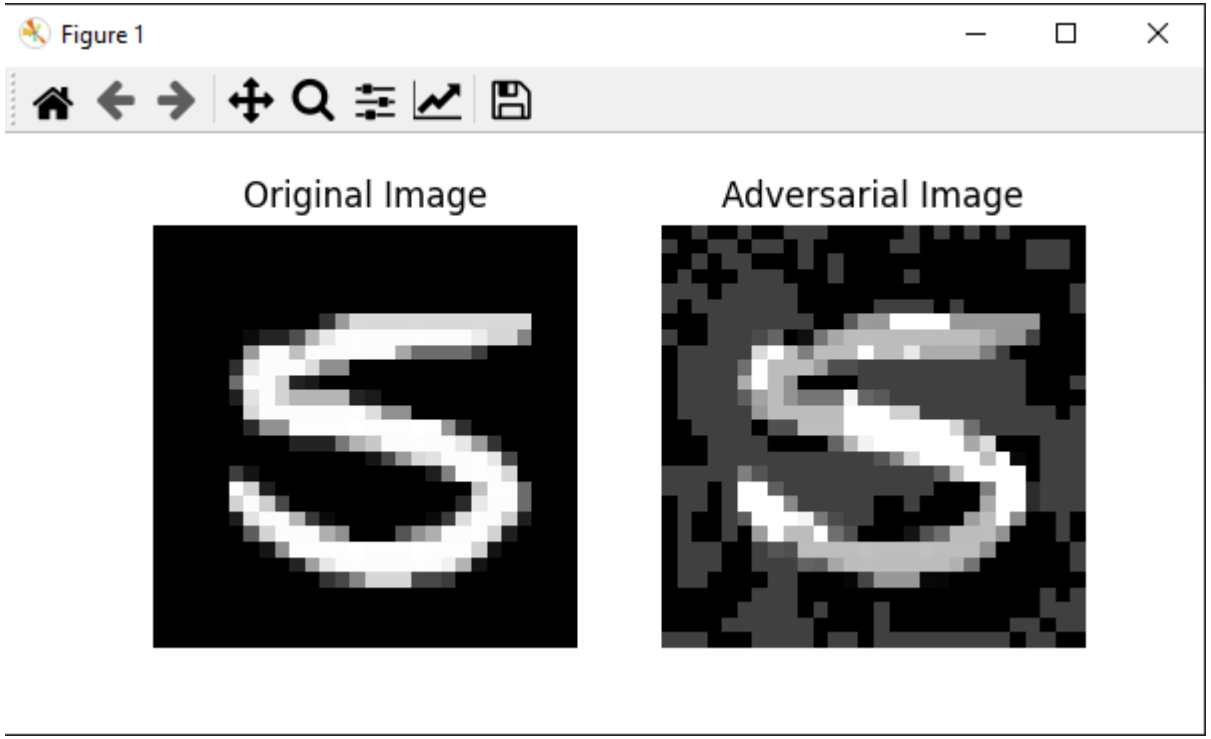


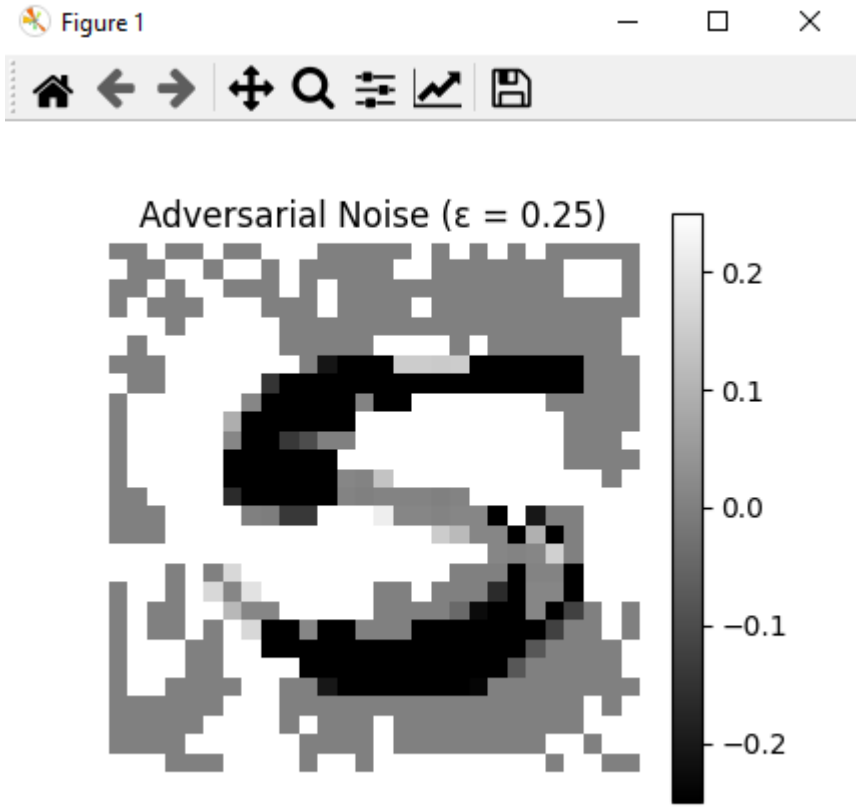
Adversarial Image

```
Gerçek etiket: 9
Model tahmini: 9
Normal tahmin: 9
Adversarial tahmin: 7
```

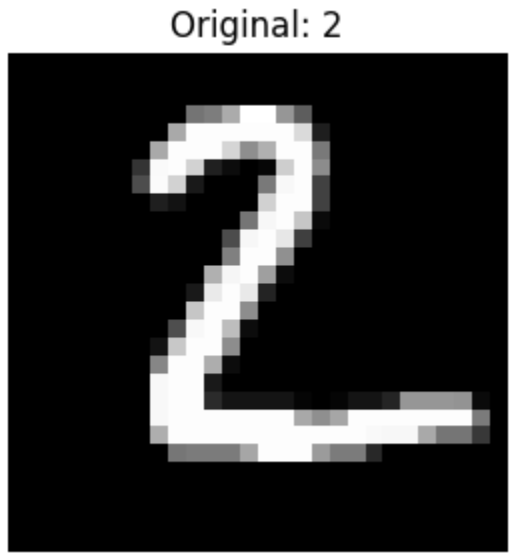
Adversarial Noise (ε = 0.25)

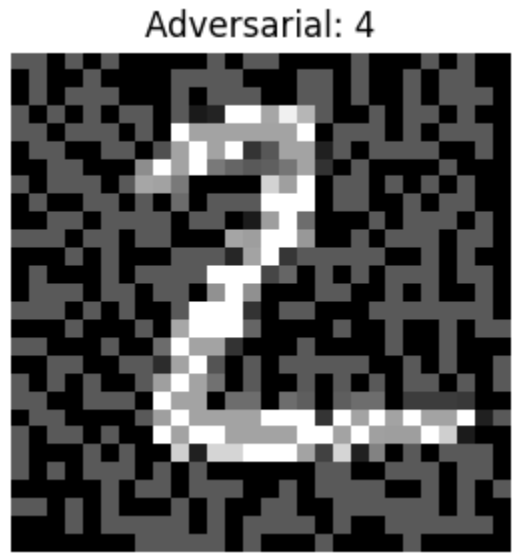


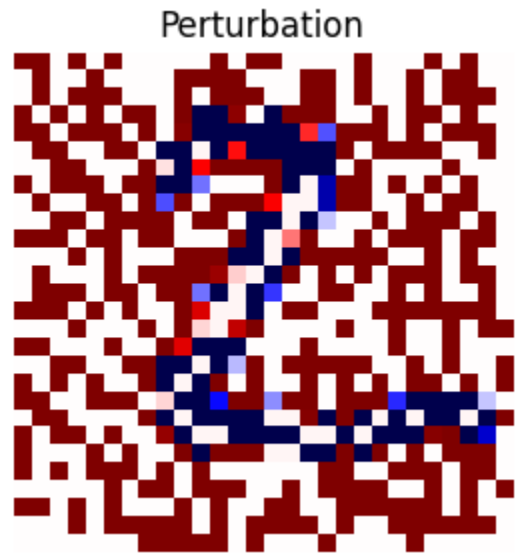


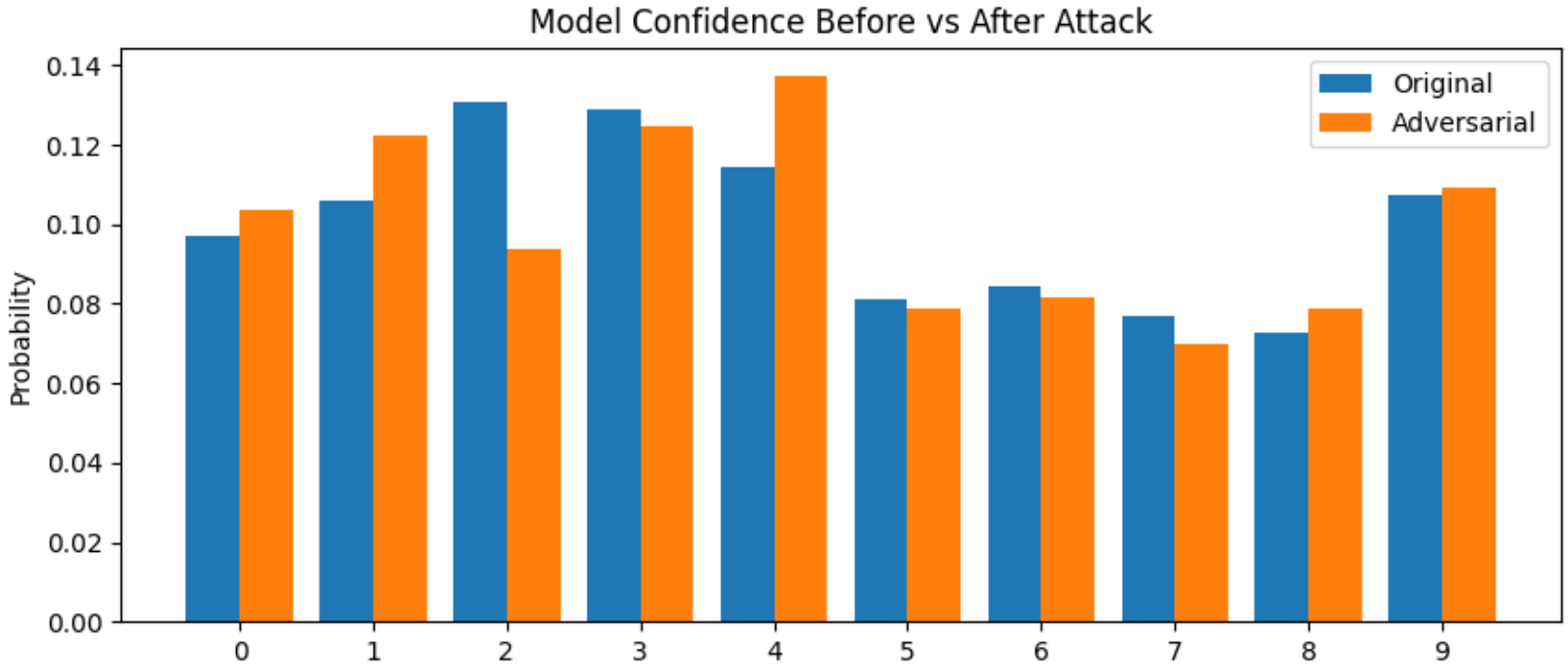


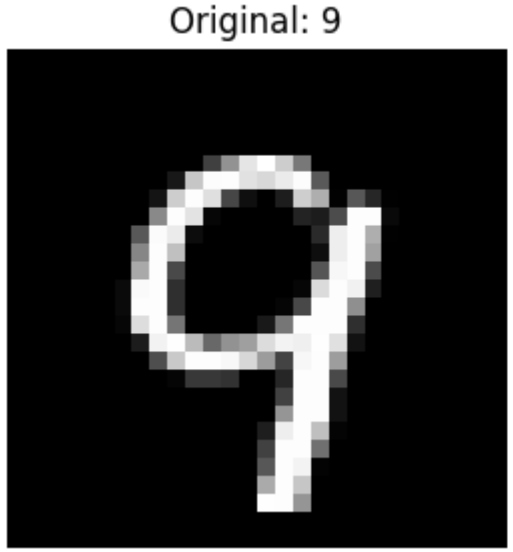

```
Gerçek etiket: 5
Model tahmini: 5
Normal tahmin: 5
Adversarial tahmin: 8
```


Original: 2


Adversarial: 4

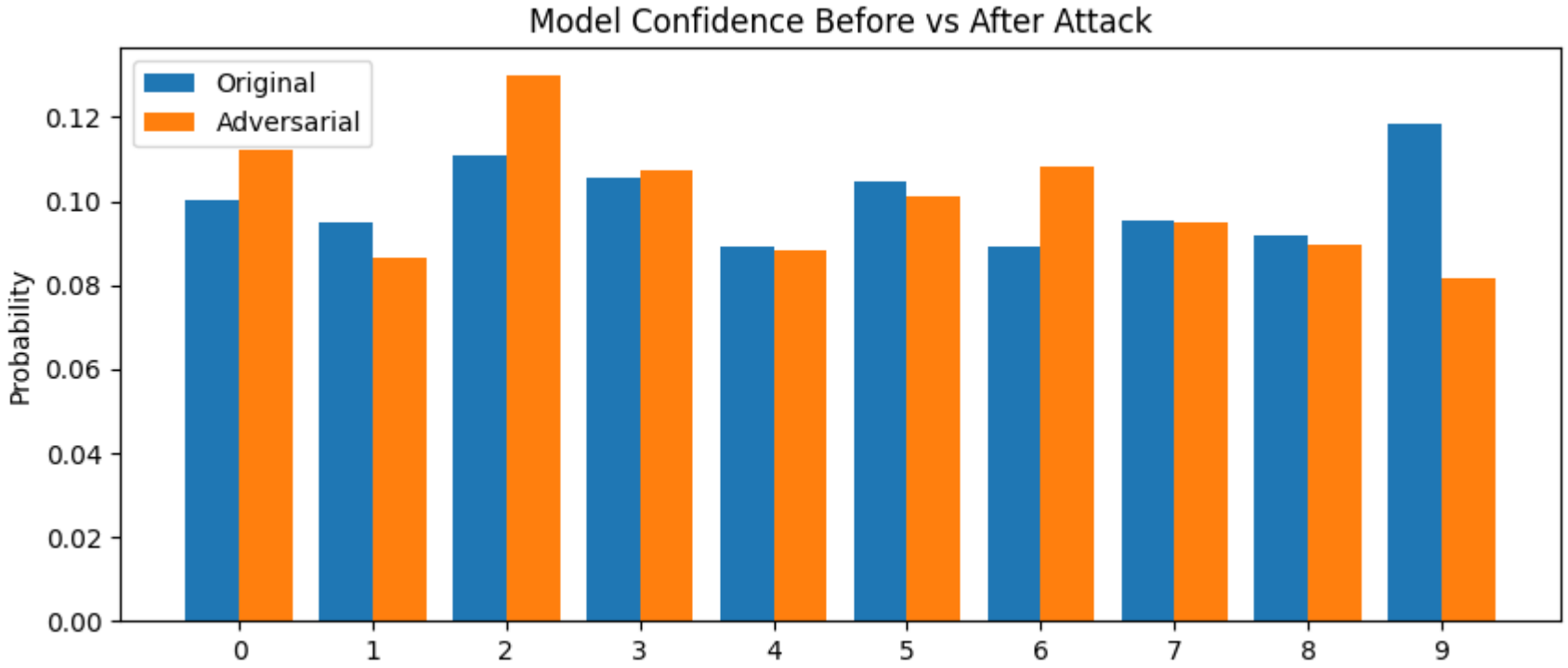

Perturbation


Model Confidence Before vs After Attack
Original
Adversarial
Probability
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
1
2
3
4
5
6
7
8
9


Original: 9


Adversarial: 2


Perturbation


Model Confidence Before vs After Attack
Original
Adversarial
Probability
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0
1
2
3
4
5
6
7
8
9

Yusuf Korkmazyiğit, Aralık 2025