İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi
Istanbul Sabahattin Zaim University Journal of the Institute of Science and Technology
https://dergipark.org.tr/tr/pub/izufbed

Derin Öğrenme Tabanlı Antivirüs Modellerinin Açık Kaynak Kodlu Çekişmeli Atak Kütüphaneleri Kullanılarak Atlatılması

Fatih ERDOĞAN[a*], Mert Can ALICI[b]

[a] *Trendyol Grup, İstanbul, Türkiye*
[b] *Prisma CSI, Ankara, Türkiye*

İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi (2021) 3 (1): 66-71
https://doi.org/10.47769/izufbed.879611

[1] 0000-0002-2075-1413; [2] 0000-0002-4553-5872

YAYIN BİLGİSİ

Yayın geçmişi:
Gönderilen tarih: 13 Şubat 2021
Kabul tarihi: 26 Şubat 2021

Anahtar kelimeler:
Derin Öğrenme
Çekişmeli Atak
Antivirüs Atlatma
Zararlı Yazılım Tespiti

ÖZET

Gelişen teknoloji ve internetin gelişmesi ile birlikte; insanların bu gelişen teknolojiyi kullanım oranının artması, kullanıcıları ve sistemlerini siber saldırganlar tarafından hedef haline getirmektedir. Siber saldırganlar tarafından kullanılan en etkili atak yöntemlerinden biri zararlı yazılımlardır. Zararlı yazılımlar aracılığı ile kişi ve kurumlara ait sistemler ele geçirilebilir, farklı enfeksiyonlara sebep olunarak daha büyük çaplı ataklar gerçekleştirilebilir. Bu saldırılar karşısında siber güvenlik firmaları tarafından geliştirilen yapay zeka tabanlı son jenerasyon antivirüs yazılımlarının yüzde yüz başarılı olamadığı görülmektedir. Gerçekleştirilen ilgili çalışmada; zararlı yazılım ve zararlı ofansif araçlara uygulanacak çekişmeli(adversarial) ataklar sonucunda üretilecek çekişmeli örnekler sayesinde, geliştirilen yapay zeka tabanlı son jenerasyon güvenlik ürünlerinin başarılı bir şekilde atlatılabildiği gözlemlenmiştir.

Overcoming Deep Learning Based Antivirus Models Using Open Source Adversarial Attack Libraries

ARTICLE INFO

Article history:
Received: 13 February 2021
Accepted:26 February 2021

Key words:
Deep Learning
Adversarial Attack
Antivirus Evasion
Malware Detection

ABSTRACT

One of the most popular attacks among cybercriminals becomes malware and its derivatives due to the recent improvements in the technology and percentage of the population that has internet access. The cybercriminals could take the control of individual or corporate systems, those attacks usually aim to keep persistence in the system for a long time. Cybersecurity companies have been developing multiple ways to detect those attacks, the latest one is an artificial intelligence-based detection engine. However, those engines cannot prevent attacks 100%. In this work, it is proved that a neural network-based malware detection engine can be bypassed by various adversarial attacks that have been prepared for the model.

1. Giriş

Günümüzde siber saldırıların artması ile kişi ve kurumlar tarafından antivirüs yazılımlarına olan talep ve kullanım oranı artmaktadır. Fakat siber saldırganlar tarafından antivirüs uygulamaları atlatılabilmektedir. Yapay zeka alanındaki ilerlemelerle birlikte antivirüs firmaları, yapay zeka tabanlı analiz motorları geliştirmeye başlamıştır. Antivirüs firmaları tarafından geliştirilen bu modeller, siber güvenlik literatüründe "Son Jenerasyon Antivirüs" olarak yerini almaktadır. Yapay zeka tabanlı analiz ve tespit modelleri, siber saldırganlar tarafından hedef haline gelmeye başlamıştır. Önceki çalışmalarda modellere saldırmak için JSMA (Rey, 2018) yöntemi uyarlandığı görülmüştür. Zararlı PDF dosyalarının tespitine yönelik gerçekleştirilen çekişmeli(adversarial) ataklarda, araştırmacılar tarafından ilgili tekniklerle PDF yapısına müdahale edilerek, PDFrate (Sukanta, 2019) ve Hidost (Srndic, 2013) alanlarından kurtulup başarılı bir şekilde zararlı yazılım sınıflandırıcısının atlatılabildiği ilgili çalışmada kanıtlanmıştır (Weilin, 2016). Windows PE dosyalarına yönelik gerçekleştirilen başka bir araştırmada alan adı üretme algoritmalarının tespitinin atlatılması amacıyla çekişmeli üretici ağlar(generative adversarial network) tabanlı çözümler geliştirildiği

görülmüştür. Zararlı yazılım örnekleri oluşturmak ve kara-kutu (black-box) tespit yöntemlerini atlatmak için kullanılan MalGan (Weiwei, 2017), çekişmeli atak uygulanması amacıyla hedef olarak seçilmiştir.

2. Çekişmeli Ataklar

Genel olarak çekişmeli ataklar, modellerin hatalı bir şekilde skor üretmesine sebep olmak için tasarlanmış girdi verileridir. Derin sinir ağları, son zamanlarda çekişmeli örnekler olarak adlandırılan iyi tasarlanmış girdi verilerine karşı savunmasız bulunmuştur (Naveed, 2018). Derin öğrenmedeki son gelişmeler, özellikle bilgisayar görmesi alanında denetimli öğrenme konusunda yoğunlaşmıştır. Bu nedenle birçok çekişmeli atak örneği, bilgisayar görmesi modellerine karşı üretilmektedir (Gamaleldin, 2018).
Örnek bir çekişmeli atak senaryosunda, bir trafik işareti tanıma sisteminde dur işareti manipüle edilerek oluşturulan çekişmeli örnek ile otonom araçların yapay zeka modellerinin yanlış sonuç üretmesine sebep olunmuştur (Kevin, 2018).

Şekil 1. Trafik işaretlerine çekişmeli atak uygulanması (Kevin E.,2018).

Bir görüntü sınıflandırma probleminde eğitilmiş bir görüntü sınıflandırıcı kullanılarak; bir kullanıcı, sınıf etiketini görüntü girdisi ile elde eder. Çekişmeli görüntüler, genellikle insanlar tarafından kolaylıkla tanınmayan, küçük bozukluklar içeren orijinal temiz görüntülerin manipüle edilmiş varyasyonlarıdır. Ancak bu tür manipülasyonlar görüntü sınıflandırıcıyı yanlış yönlendirmektedir.
Bunun sonucunda kullanıcı, sınıflandırıcıya girdi olarak verdiği görüntü için yanlış bir skor ya da yanlış bir sınıf sonucunu elde eder.

2.1 Beyaz-Kutu (White-Box) Atak

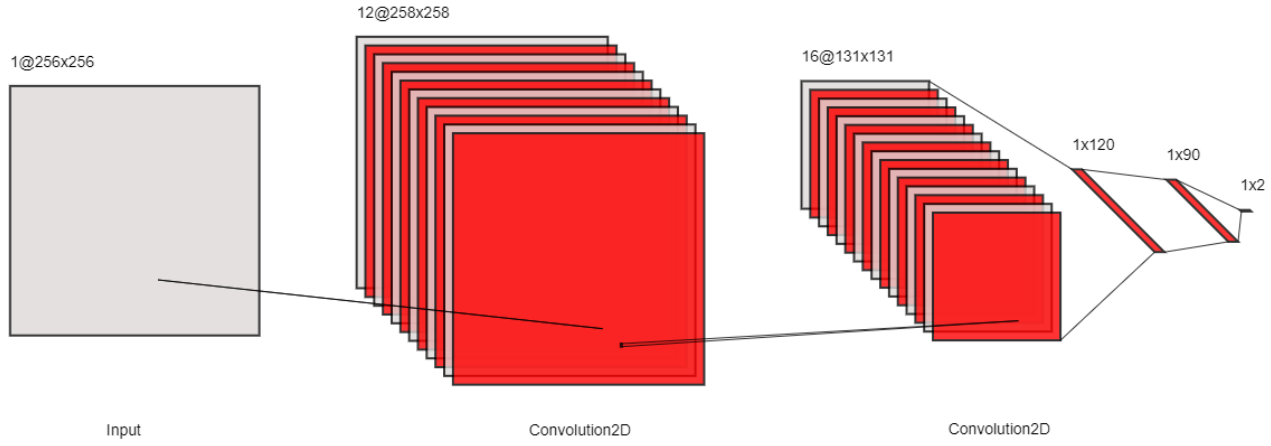
Beyaz-kutu (white-box) ataklar model ve eğitim veri seti hakkında uçtan uca bilgi sahibi olunarak yapılan ataklardır. Hedeflenen modele yönelik; eğitilmiş sinir ağı modelleri, eğitim veri seti, model mimarileri, hiper parametreler, katman sayıları, aktivasyon fonksiyonları, model ağırlıkları vb. gibi bilgilere sahip olunarak çekişmeli ataklar gerçekleştirilmektedir (Xiaoyong, 2019).
Model gradyanları hesaplanarak birçok çekişmeli örnek oluşturulmaktadır. Derin öğrenme tabanlı yapay sinir ağları genellikle el yordamıyla elde edilen öznitelikler yerine ham veri üzerinde çalıştığından, öznitelik seçimi makine öğrenimindeki çekişmeli atak örneklerine göre gerekli değildir.

2.2 Kara-Kutu (Black-Box) Atak

Kara-kutu(black-box) ataklar, eğitilmiş sinir ağı modeli hakkında hiçbir bilgi sahibi olunmadan gerçekleştirilen ataklardır. Saldırgan, standart bir kullanıcı olarak hareket eder ve yalnızca modelin skoru ya da sınıfı hakkında bilgi sahibidir. (Xiaoyong Y.,2019).
Bu atak yöntemi kullanılarak AWS (Krizhevsky, 2012), Google Cloud AI (Simonyan, 2014), BigML (Redmon, 2016), Clarifai (Ren, 2015), Microsoft Azure (Saon, 2015), IBM Bluemix (Sutskever, 2014), Face++ (Oord, 2016) gibi çevrimiçi makine öğrenimi hizmetlerine ataklar gerçekleştirilmektedir.

3. Son Jenerasyon Antivirüs

Geleneksel antivirüslerin ilk birkaç ya birkaç milyon kullanıcıda zararlı yazılım sinyallerinin alınması üzerine zararlı yazılım analistleri, bu zararlı yazılımlardan imza üreterek imza veritabanlarına eklemektedir.
Bunun haricinde otomatik analiz ve tespit imzaları üretme mekanizmaları da kullanılmaktadır. Bundan sonra bu güncellemeyi alan kullanıcılar bu zararlı yazılımların cihazlarını enfekte etmesinden korunabilecektir. Ancak sinyallerini aldığımız ilk birkaç ya da birkaç milyon kullanıcı enfekte olacaktır. Fakat son jenerasyon antivirüslerde bunun aksine istatistiksel modeller kullanılarak bir yazılımın zararlı olup olmadığına karar verilir.
Makine öğrenmesi ya da derin öğrenme methodları kullanılarak oluşturulan antivirüs çözümleri, genelleme yetenekleri sayesinde sadece bir örnek ya da bir aile değil, zararlı yazılım davranışlarını öğrenerek kurban vermeden zararlı yazılımları tespit etmeyi amaçlar.



Şekil 2. Derin öğrenme tabanlı zararlı yazılım sınıflandırma modeli.

4. Derin Öğrenme Tabanlı Zararlı Yazılım Tespit Modeli

Şekil 2'de görüldüğü üzere; geliştirilen derin öğrenme tabanlı zararlı yazılım sınıflandırma modelinde, bilgisayar görmesi alanında da kullanılan Convolutional Neural Network (CNN) (Yann, 1995) katman yapısı baz alınmaktadır.

Bu baz katmanın çalıştırılabilmesi için girdi olarak 1 veya 3 kanallı bir görüntü verisine ihtiyaç duyulmaktadır.

Geliştirilen model, Virustotal servisi kullanılarak bir milyon adet zararlı yazılım ve bir milyon adet temiz dosya veri seti ile eğitilmiştir. Bu veri setinde zararlı yazılım aileleri ve zararlı yazılım davranışlarına göre eşit sayıda örnek bulunmaktadır. Eğitim süresince Tensorflow/Keras kütüphanesi, bir adet Nvidia RTX 2080Ti ekran kartında ve 16 çekirdek AMD Ryzen 5 donanımı kullanılmıştır.

Yapay zeka modelinin geliştirilmesi ve eğitilmesi aşamalarında kullanılan zararlı yazılım örnekleri (1, 256, 256) boyutlu görüntü dizilerine dönüştürülerek ilgili modelin girdisi olmaktadır.

- Conv2D katmanından geçerek ilk öznitelikler çıkartılır.
- MaxPool2D (Alessandro, 2013) katmanında süzgeçten geçirilir.
- Conv2D katmanından geçerek zararlı yazılım hakkında daha detaylı öznitelikler çıkartılır.
- MaxPool2D katmanında tekrar süzgeçten geçirilir.
- Daha sonra elde edilen çok boyutlu veri, tek boyuta çevrilir. Bu işleme Flatten adı verilmektedir.
- En son bulunan üç adet FullyConnected katmanıyla, veri hem daraltılmaktadır hem de istediğimiz etiketlere dönüştürülmektedir.

Belirtilen model, çıktı olarak iki skor değeri bildirir. Bu skor değerlerinin toplamı 1'e eşittir. Yüksek olan skor değeri, örneğin hangi sınıfa dahil olduğunu belirler.

Şekil 3'te görüldüğü üzere veri setinde yer alan dosyalara ait skorlar elde edilmektedir.

```
/malware_image/azorult.png          :   CLEAN     0.10591787844896317
/malware_image/rev-shell-32-msf.png       :   MALICIOUS 0.7929524779319763
/malware_image/rev-vnc-32-msf.png         :   MALICIOUS 0.6748470664024353
/malware_image/rev-meter-32-msf.png       :   MALICIOUS 0.7725298404693604
/malware_image/trickbot_1.png       :   MALICIOUS 0.9998303651809692
/malware_image/mimilove.png         :   MALICIOUS 0.5499293208122253
/malware_image/kpotstealer.png      :   CLEAN     0.020229043439030647
/malware_image/turkojan.png         :   MALICIOUS 0.9966475367546082
/malware_image/bitter_rat.png       :   CLEAN     0.006656558718532324
/malware_image/grandcab.png         :   MALICIOUS 0.9737184643745422
/malware_image/hfs.png              :   MALICIOUS 0.803017795085907
```

Şekil 3. Zararlı yazılım tespit skorlarının elde edilmesi.

5. Atak Veri Seti Hazırlığı

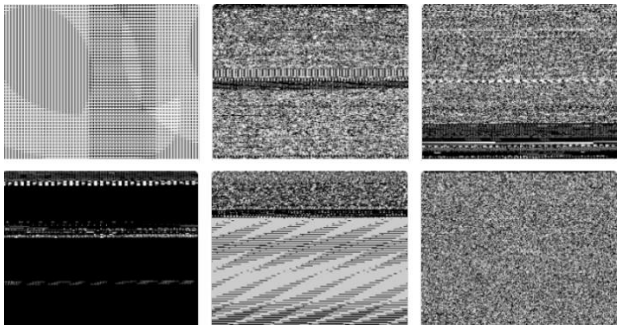
İlk olarak çekişmeli atak uygulanacak olan zararlı yazılımlar ve ofansif amaçla kullanılan güvenlik araçları toplanmıştır.

Çizelge 1. Zararlı yazılım atak veri seti

Zararlı yazılım çeşitleri	Sayı
Nanocore	25
Lokibot	23
AgentTesla	8
Trickbot	4
Mimikatz	4
Ryuk Ransomware	3
Metasploit	3
Gh0st Rat	2
Artra Downloader	2
Gandcrab Ransomware	2
Dridex	2
TA-505	1
Qakbot	1
Pushdo	1
Pony Rat	1
Shade Ransomware	1
Turkojan	1
Zeus	1
Autolt Malware	1
AveMaria Rat	1
Bitter Rat	1
Nanobot	1
Kovter	1
Kpot Stealer	1
Formbook	1
Azorult Stealer	1
Calypso Dropper	1
CoinMiner	1
beRoot	1
hfs	1

.

Çizelge 1'de görüldüğü üzere farklı ailelere ait 100 adet zararlı yazılım, tehdit istihbaratı yöntemleriyle elde edilmiştir. Modele girdi olarak verilen 100 adet zararlı yazılımın 60 tanesinin geliştirilen model tarafından tespit edildiği görülmüştür. Geriye kalan 40 adet zararlı yazılım örneği true-negative olarak değerlendirilmiştir.



Şekil 4. Hedef uygulamaların modele uygun hale getirilmesi.

Başarılı bir şekilde tespit skoru almak için hedef zararlı uygulamaların resme dönüştürülüp yapay zeka modeline girdi olarak verilmesi gerekmektedir. Ardından girdi olarak verilen exe formatındaki her dosya, Şekil 4'teki gibi resme dönüştürülüp png formatında kaydedilmektedir.

Bu aşama sonucunda hedef olarak seçilen zararlı yazılım örnekleri, geliştirilen yapay zeka modeline girdi olarak verilip başarılı bir şekilde skor elde edilebilir hale gelir.

6. Adversarial Robustness Toolbox - ART

Adversarial Robustness Toolbox (ART) (Maria-Irina N.,2018), geliştiricilerin ve araştırmacıların makine öğrenimi modellerini ve uygulamalarına karşı çekişmeli ataklar geliştirmelerine, bu ataklara karşı savunma mekanizmaları oluşturmaya yarayan araçtır. Hazırlanan veri seti üzerinde çekişmeli ataklar gerçekleştirmek amacıyla, açık kaynak

kodlu ve IBM tarafından geliştirilen ART birçok çekişmeli atak methodunu içerisinde barındırmaktadır.

7. Çekişmeli Atakların Belirlenmesi

Çekişmeli atakların önem kazanmasıyla birlikte araştırmacılar tarafından farklı methodlar geliştirilmiştir. Geliştirilen bu methodların birçoğunu ART, bünyesinde barındırmaktadır. ART içerisinde bulunan ve geliştirilen yapay zeka modeli üzerinde çalışabilecek atakların listesi aşağıdaki gibidir.

- FGSM Attack (Goodfellow I.,2015)
- Adversarial Patch (Brown T.,2017)
- Boundary Attack (Brendel W.,2017)
- Carlini and Wagner L2 Attack (Carlini N.,2017)
- Carlini and Wagner Linf Attack (Carlini N.,2017)
- DeepFool (Dezfooli S.,2015)
- Elastic Net Attack (Chen P.,2017)
- Frame Saliency Attack (Nathan I.,2018)
- HopSkipJump Attack (Jianbo C.,2020)
- Basic Iterative Method (Alexey K.,2016)
- Projected Gradient Descent (Madry A.,2017)
- NewtonFool (Uyeong J.,2017)
- Jacobian Saliency Map Attack (Nicolas P.,2016)
- Shadow Attack (Amin G.,2020)
- Spatial Transformations Attack (Logan E.,2019)
- Square Attack (Andriushchenko M.,2020)
- Universal Perturbation Attack (Dezfooli M.,2017)

8. Çekişmeli Atakların Uygulanması

ART içerisinde bulunan çekişmeli ataklar başarılı bir şekilde implemente edildikten sonra, atakları gerçekleştirecek ve raporlama yapacak fonksiyonlar Şekil 5'te gösterildiği gibi tasarlanmıştır.

Temel olarak her bir atağın oluşturulup, oluşturulan atakların örnek zararlı yazılım üstünde uygulanması sonucu elde edilen çekişmeli atak görselleri üretilerek gerçekleştirilir. Bu aşama sonucunda çekişmeli ataklar başarılı bir şekilde gerçekleştirilip sonuçları kaydedilir.

Uygulanan çekişmeli atağın başarı durumunun belirlenmesi için α eşik değeri 0.5 olarak kabul edilmiştir.

Algorithm 1: Çekişmeli atakların uygulanması

```
Result: Çekişmeli atak uygulanmış örnek
resim içeriğini dosyadan oku;
resim kanalını 1 yap;
resmi 256x256 boyutlandır;
modelden orijinal tespit skoru üret;
for atak methodları do
    atak oluştur;
    örnek üzerinde atağı uygula;
    modelden atak skorunu elde et;
    if atak skoru > α then
        başarılı atak;
    else
        başarısız atak;
    end
    atak skoru orijinal skoru karşılaştır;
end
```

Şekil 5. Çekişmeli atakların uygulanması.

Uygulanan her bir çekişmeli atağın başarı durumuna, skor değerine ve değişim oranına Şekil 6'te yer verilmiştir.

```
[/content/drive/My Drive/AV_RESEARCH/samples/malware_image/turkojan.png]
Attack Name                          Prediction        Score        Change  Success
-----------------------------------  ------------  -----------  ------------  ---------
attack_fast_gradient_method            0.996648  2.32332e-08   0.996648     True
attack_adversarial_patch               0.996648  0.996206      0.000441849  False
attack_boundary_attack                 0.996648  0.996648      0            False
attack_carliniL2method                 0.996648  0.417274      0.579373     True
attack_carliniLInfmethod               0.996648  0.176475      0.820172     True
attack_deep_fool                       0.996648  0.372974      0.623673     True
attack_elasticnet                      0.996648  0.451067      0.545581     True
attack_frame_saliency_attack           0.996648  6.28682e-10   0.996648     True
attack_hopskipjump                     0.996648  0.996648      0            False
attack_basic_iterative_method          0.996648  1.29157e-24   0.996648     True
attack_projected_gradient_descent      0.996648  1.29157e-24   0.996648     True
attack_newton_fool                     0.996648  1            -0.00335246   False
attack_saliency_mapmethod              0.996648  0.996648      0            False
attack_shadow_attack                   0.996648  0.996748     -0.000100434  False
attack_spatial_transformation          0.996648  0.996648      0            False
attack_square_attack                   0.996648  0.440499      0.556149     True
attack_universal_perturbation          0.996648  0.372974      0.623673     True

Success / Total
10 / 17
```

Şekil 6. Çekişmeli atak sonuçları.

9. Sonuç

Bu çalışmada geliştirilen yapay zeka modeli ile veri seti içerisinden zararlı yazılım olarak tespit edilen 60 adet dosyaya, 17 adet çekişmeli atak uygulanmıştır ve sonuçlar incelenmiştir.

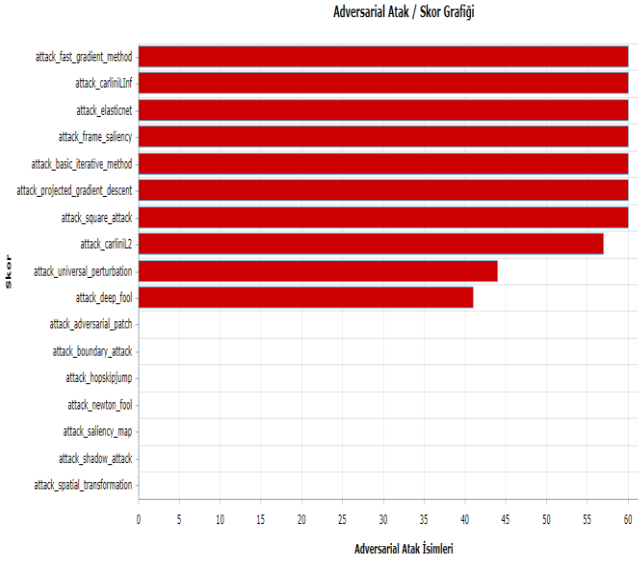
İlgili yapay zeka modelinin geliştirilmesi ve çekişmeli atakların gerçekleştirilmesi beyaz-kutu(white-box) atak kapsamına girmektedir.

İlk olarak uygulanan 17 adet çekişmeli ataktan kaç tane zararlı yazılım örneğinde başarılı oldukları ve başarılı olan ataklar Çizelge 2'de gösterilmiştir.

Çizelge 2. Çekişmeli atak-başarı çizelgesi

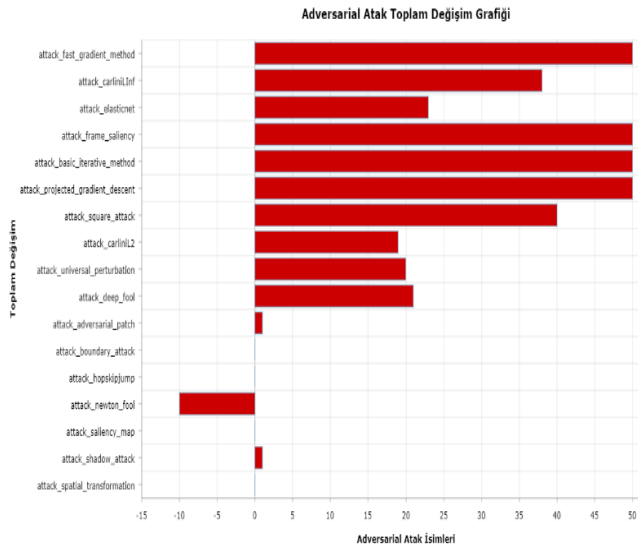
Başarılı olan çekişmeli ataklar	***Sayı***
Fast Gradient (FGSM) Attack	60
Carlini and Wagner Linf Attack	60
Elastic Net Attack	60
Frame Saliency Attack	60
Basic Iterative Method	60
Projected Gradient Descent	60
Square Attack	60
Carlini and Wagner L2 Attack	57
Universal Perturbation Attack	43
DeepFool	41

Çekişmeli ataklar sonucu gerçekleşen skor değişimleri incelenerek, en başarılı ve en başarısız çekişmeli ataklar Şekil 7'de görülmektedir.



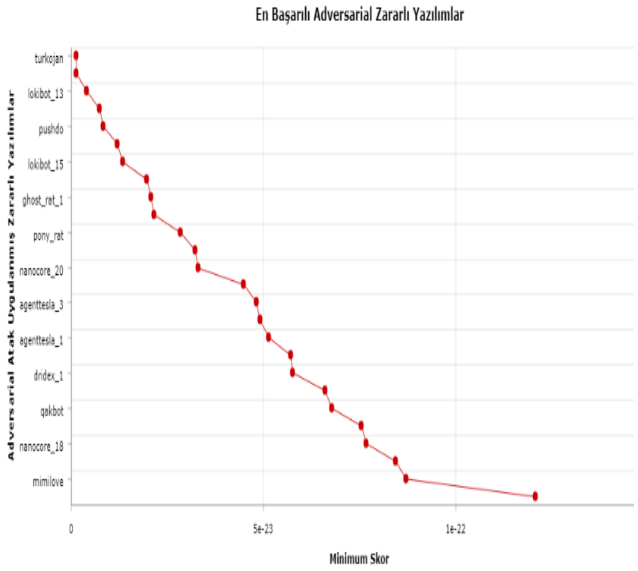
Şekil 7. Çekişmeli atak-başarı grafiği.

Başarı değerine göre en yüksek değerle başarılı olan çekişmeli atakların sıralaması Şekil 8'deki gibidir.



Şekil 8. Çekişmeli atakların skor değişim grafiği.

En başarısız olan çekişmeli atağın "NewtonFool" atak yöntemi olduğu tespit edilmiştir.
Çekişmeli atak uygulanan zararlı yazılım örnekleri arasından, geliştirilen yapay zeka modelini atlatma konusunda skorları toplamda en çok değişen ve en az değişen zararlı yazılımlara Şekil 9'de yer verilmiştir.
Şekil 9'de görüldüğü üzere en başarılı çekişmeli zararlı yazılım "turkojan" olarak tespit edilmiştir. Turkojan zararlı yazılımı, Türk saldırganlar tarafından geliştirilmiş bir truva atı zararlı yazılımıdır.
Gerçekleştirilen proje ve araştırma sonucunda; zararlı yazılım ve zararlı ofansif araçlara uygulanacak doğru çekişmeli ataklar sonucunda üretilecek çekişmeli örnekler sayesinde, geliştirilen son jenerasyon güvenlik ürünlerinin ve yapay zeka modellerinin başarılı bir şekilde atlatılabildiği kanıtlanmıştır.



Şekil 9. En başarılı çekişmeli atak uygulanmış zararlı yazılım.

10. Gelecek Çalışmalar

Bundan sonraki çalışmalarda çekişmeli atak uygulanmış zararlı yazılım örneklerinden, orijinal çalıştırılabilir formattaki dosyaların ön işleme fonksiyonundan geçtikten sonra bu örnekleri çıktı verebilir hale dönüştürülüp ardından tespit motorlarının yanlış sonuç vermesi sağlanabilir.
Yapılan çalışmada bilgisayar görmesi alanındaki methodlar kullanılarak oluşturulmuş bir tespit modeli kullanılmıştır. Devamı olarak, doğal dil işleme yöntemleri aracılığıyla oluşturulan ve zaman ekseninde (baytlar üzerinde) veri işleyebilen modellere de ataklar tasarlanabilir.
Çekişmeli üretici ağlar kullanılarak iki yapay sinir ağının birbiri ile çekişmesi sonucu zararlı yazılım örneklerinden çekişmeli atak örnekleri üretilebilir. Buradaki ayrıştırıcı yapay sinir ağında bir tespit modeli ya da bir tespit ürünü kullanılabilir.
Planlanan gelecek çalışmalar neticesinde en başarılı atak senaryoları belirlenip, siber güvenlik ekosisteminde yer alan son jenerasyon antivirüs ürünlerine yönelik kara-kutu(black-box) ataklar hazırlanabilir.

Kaynaklar

Rey, W., Xu, A. (2018). Maximal jacobian-based saliency map attack. arXiv preprint arXiv:1808.07945.

Sukanta, D. et al. (2019). EvadePDF: Towards Evading Machine Learning Based PDF Malware Classifiers. International Conference on Security and Privacy. Springer, Singapore.

Srndic, N., Laskov, P. (2013). Detection of Malicious Pdf Files Based on Hierarchical Document Structure. In 20th Network and Distributed System Security Symposium (NDSS).

Weilin, X., Yanjun, Q., Evans D. (2016). Automatically evading classifiers. Proceedings of the 2016 network and distributed systems symposium. Vol. 10.

Weiwei, H., Tan, Y. (2017). Generating adversarial malware examples for black-box attacks based on gan. arXiv

preprint arXiv:1702.05983.

Naveed, A., Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access 6: 14410-14430.

Gamaleldin, E. et al. (2018). Adversarial examples that fool both computer vision and time-limited humans. Advances in Neural Information Processing Systems.

Kevin, E. et al. (2018) Robust physical-world attacks on deep learning visual classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Xiaoyong, Y. et al. (2019). Adversarial examples: Attacks and defenses for deep learning. IEEE transactions on neural networks and learning systems 30.9: 2805-2824.

Krizhevsky, A., Sutskever, I., Hinton, G., E. (2012), Imagenet classification with deep convolutional neural networks, in Advances in neural information processing systems, pp. 1097–1105.

Simonyan, K., Zisserman, A. (2014), Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

Redmon, J., Farhadi, A. (2016), Yolo9000: Better, faster, stronger, arXiv preprint arXiv:1612.08242.

Ren, S., He, K., Girshick, R., Sun, J. (2015), Faster r-cnn: Towards realtime object detection with region proposal networks, in Advances in neural information processing systems, pp. 91–99.

Saon, G., Kuo, J., Rennie, S., Picheny, M. (2015), The ibm 2015 english conversational telephone speech recognition system, arXiv preprint arXiv:1505.05899.

Sutskever, I., Vinyals O., Le, V. (2014), Sequence to sequence learning with neural networks, in Advances in neural information processing systems, pp. 3104–3112.

Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, A., Senior, A., Kavukcuoglu, K. Wavenet (2016): A generative model for raw audio, arXiv preprint arXiv:1609.03499.

Yann, L., Bengio, Y. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361.10.

Alessandro, G. et al. (2013). Fast image scanning with deep max-pooling convolutional neural networks. 2013 IEEE International Conference on Image Processing. IEEE.

Maria-Irina, N. et al. (2018) Adversarial Robustness Toolbox v1. 0.0. arXiv preprint arXiv:1807.01069.

Goodfellow, I., Shlens, J., Szegedy, C. (2015). Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR).

Brown, T., Mane, D., Roy, A., Abadi, M., Gilmer, J. (2017). Adversarial patch. CoRR, abs/1712.09665.

Brendel, W., Rauber, J., Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. CoRR, 1712.04248.

Carlini, N., Wagner, D. (2017). Towards evaluating the robustness of neural networks. In IEEE Symposium on Security and Privacy.

Dezfooli, S., Fawzi, A., Frossard, P. (2015). Deepfool: a simple and accurate method to fool deep neural networks. CoRR, abs/1511.04599.

Chen, P., Sharma, Y., Zhang, H., Yi, J., Hsieh, E. (2017): Elastic-net attacks to deep neural networks via adversarial examples. CoRR, abs/1709.04114.

Nathan, I. et al. (2018). Adversarial attacks for optical flow-based action recognition classifiers. arXiv preprint arXiv:1811.11875.

Jianbo, C., Jordan, M., Wainwright, M. (2020). Hopskipjumpattack: A query-efficient decision-based attack. 2020 ieee symposium on security and privacy (sp). IEEE.

Alexey, K., Goodfellow, I., Bengio, S. (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.

Madry, A. et al. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

Uyeong, J., Wu, X., Jha, S. (2017). Objective metrics and gradient descent algorithms for adversarial examples in machine learning. Proceedings of the 33rd Annual Computer Security Applications Conference.

Nicolas, P. et al. (2016). The limitations of deep learning in adversarial settings. 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE.

Amin, G., Shafahi, A., Goldstein, T (2020). Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. arXiv preprint arXiv:2003.08937.

Logan, E. et al. (2019) Exploring the landscape of spatial robustness. International Conference on Machine Learning. PMLR.

Andriushchenko, M. et al. (2020). Square attack: a query-efficient black-box adversarial attack via random search. European Conference on Computer Vision. Springer, Cham.

Dezfooli, M., Mohsen, S. et al. (2017) Universal adversarial perturbations. Proceedings of the IEEE conference on computer vision and pattern recognition.