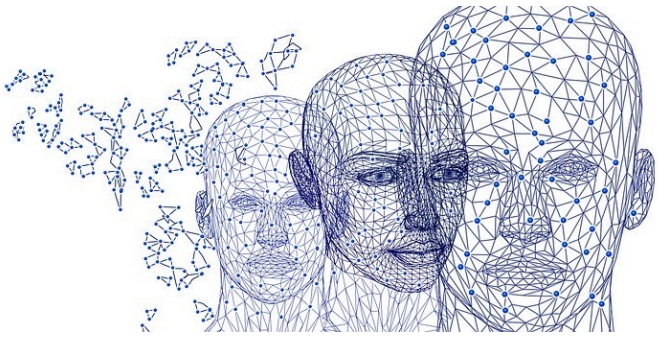
February 2021, Issue Code NL-2021-4

DDD (Digital Data Deception) Technology Watch Newsletter

Table of Contents



"*All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near.*"

— Sun Tzu, *The Art of War*

Editors: Enes Altuncu, Virginia Franqueira, Sanjay Bhattacherjee and Shujun Li
Affiliation: Kent Interdisciplinary Research Centre in Cyber Security (KirCCS), University of Kent, UK
Contact Us: ddd-newsletter@kent.ac.uk

© 2021 University of Kent, UK

Editorial

In this fourth issue of the Digital Data Deception (DDD) Technology Watch Newsletter, we cover a range of topics relevant for DDD.

We start with "Information Hiding in Images", taking the perspective of *Image Steganography*, and *Image Steganalysis*. A new trend in this domain is the subject of "coverless image steganography" which we intend to further explore in an upcoming issue.

The second section of this newsletter reviews the topic of "Fake Software and Services" from the perspective of *Fake and Malicious Mobile Apps*, and *Fake Antivirus Software*. An example of digital deception via legitimate software made headlines in December 2020 when the network monitoring solution called Orion, produced by SolarWinds, was found to be providing unauthorised remote access to the internal network of government agencies (6 in the US) and big technology companies worldwide such as Microsoft (`https://www.theguardian.com/technology/2020/dec/15/orion-hack-solar-winds-explained-us-treasury-commerce-department`). It is now being considered a large-scale, highly sophisticated, digital espionage operation (`https://www.bbc.co.uk/news/technology-55368213`).

The last two sections of this newsletter focus on AI systems in terms of *Poisoning Attacks* and *Backdoor Attacks*, and in terms of vulnerabilities – *Security Analysis of AI Systems* and *Attacks Exploiting Vulnerabilities in AI Systems*. The power of AI for deception was made evident via the "Alternative Christmas Day Message" of 24 December 2020, broadcast by Channel 4 (UK) (`https://www.channel4.com/press/news/deepfake-queen-deliver-channel-4s-alternative-christmas-message`). The message, part of a campaign to raise awareness about Deepfake technology as a vehicle to spread misinformation and disinformation, featured a "fabricated" Queen speaking to the nation.

This issue summarises 24 research papers published since 2019, and cites another supporting paper. The search for papers followed a venue-driven systematic literature review (SLR) approach.

This issue has an addendum Chinese section (NL-2021-4-C) where the scope of the DDD technology watch is extended to research papers published in Chinese; this section is available upon request. The Chinese section for this issue covers (1) Adversarial AI, and (2) Information Hiding in different types of media.

We hope you enjoy reading this issue. Feedback is always welcome, and should be directed to ddd-newsletter@kent.ac.uk.

© 2021 University of Kent, UK

List of Acronyms

- ABS: Artificial Brain Stimulation
- AI: Artificial Intelligence
- API: Application Programming Interface
- APK: Android Application Package
- AV: Antivirus
- BBC: Block Boundary Continuity
- BBM: Block Boundary Maintenance
- CALPA-NET: Channel-Pruning-Assisted Network
- CNN: Convolutional Neural Network
- CP: Convex Polytope (Attack)
- DCT: Discrete Cosine Transform
- DL: Deep Learning
- DNN: Deep Neural Network
- DRL: Deep Reinforcement Learning
- FC: Feature Collision (Attack)
- FRAD: Fake Removal Information Advertisement
- GAN: Generative Adversarial Network
- GIF: Graphics Interchange Format
- GTSRB: German Traffic Sign Recognition Benchmark
- HOG: Histograms of Oriented Gradients
- IDLSes: Interpretable Deep Learning Systems
- IMAD: Illegitimate Mobile App Detector
- IoT: Internet of Things
- IoU: Intersection over-Union
- JPEG: Joint Photographic Experts Group
- LDA: Latent Dirichlet Allocation
- LISA: Laboratory for Intelligent & Safe Automobiles
- ML: Machine Learning
- MLaaS: Machine Learning as a Service
- MMP: Mobile Market Place
- NC: Neural Cleanse
- SCA: Side-Channel Aware
- SGAN: Steganographic Generative Adversarial Network
- TORCS: The Open Racing Car Simulator
- TLSH: Trend Micro Locality Sensitive Hash
- UCB: Upper Confidence Bound
- VGG: Visual Geometry Group

© 2021 University of Kent, UK

Information Hiding in Images

Introduction

Steganography is a major sub-domain of digital information hiding; other sub-domains are watermarking and cryptography [1]. Its purpose is to conceal the existence of hidden communication between a sender and a receiver (subjects or objects), creating *covert channels*, where the hidden data (called "payload") is embedded into digital media (called "carrier" or "cover") resulting in a "stego carrier" (or "stego-image"). The stego, containing hidden data, may be protected with a "stego key", pre-agreed between parties. A range of carrier types can be used for different applications such as image, audio, video, text, network traffic, and IoT protocols. In the last couple of years, intense development in steganography and steganalysis (aiming at attacking the robustness and detectability of steganographic techniques) has been happening, especially with the use of CNN- and GAN-based approaches. This section focuses on *image* files as a carrier.

Image Steganography

Cogranne et al. [2] have extended the so-called MiPOD scheme to design a distortion function for JPEG-compressed images that is statistically founded. The MiPOD scheme is based on minimising the detection accuracy of the most powerful test using a Gaussian model of independent DCT coefficients. This method is also applied to address the problem of hiding information in colour JPEG images. The main issue in such cases is that colour channels are not processed in the same way and, hence, a statistical approach is expected to bring significant improvements when one needs to consider heterogeneous channels together. The results presented showed that, on the one hand, the extension of MiPOD for JPEG domain (referred to as J-MiPOD) is very competitive in comparison to current state-of-the-art embedding schemes. On the other hand, they also showed that addressing the problem of hiding information in JPEG colour images is far from being straightforward. So, future works are required to understand better how to deal with colour channels in JPEG images.

Editorial Notes

The MiPOD scheme works with the assumption that each pixel follows the Gaussian distribution and is statistically independent from the others. This may not be true for real-life images. It may be interesting to explore the effect of the actual pixel distributions in this context.

Wang et al. [17] have presented results on *content-adaptive steganography* based on the minimum distortion embedding framework, which tends to embed hidden messages into textured and noisy regions, making them difficult to detect by the steganalyser. They started from the context that a reasonable non-additive cost function can significantly improve the security level of additive cost based steganography. There is only one principle so far, called *block boundary continuity* (BBC), that has been proposed to define the non-additive cost function for JPEG steganography. BBC aims to synchronise the modification direction of inter-block boundaries in the spatial domain. In this article, the authors have found that JPEG steganography usually introduces more and larger modifications on the boundary than on the inside of each intra-block in the spatial domain, which is another important factor affecting security. This led them to a new principle, called *block boundary maintenance* (BBM), to minimise the modifications on the spatial block boundaries. They have theoretically deduced the BBM principle on how to modify a pair of intra-block *discrete cosine transform* (DCT) coefficients to reduce the modifications on the spatial block boundary. They have designed a new strategy to define non-additive cost functions for JPEG steganography by exploiting the intra-block correlation coefficient in the DCT domain. Their experimental results showed that the BBM-based strategy can minimise modifications on the spatial block boundaries and thus achieve a high-security level when resisting modern JPEG steganalysis. Furthermore, the two principles of BBC and BBM can be fused to further improve the empirical security.

Qin et al. [10] provided a comprehensive survey of *coverless image steganography*, a concept first introduced in 2014. This technique hides data via the properties of the cover image such as pixel brightness

© 2021 University of Kent, UK

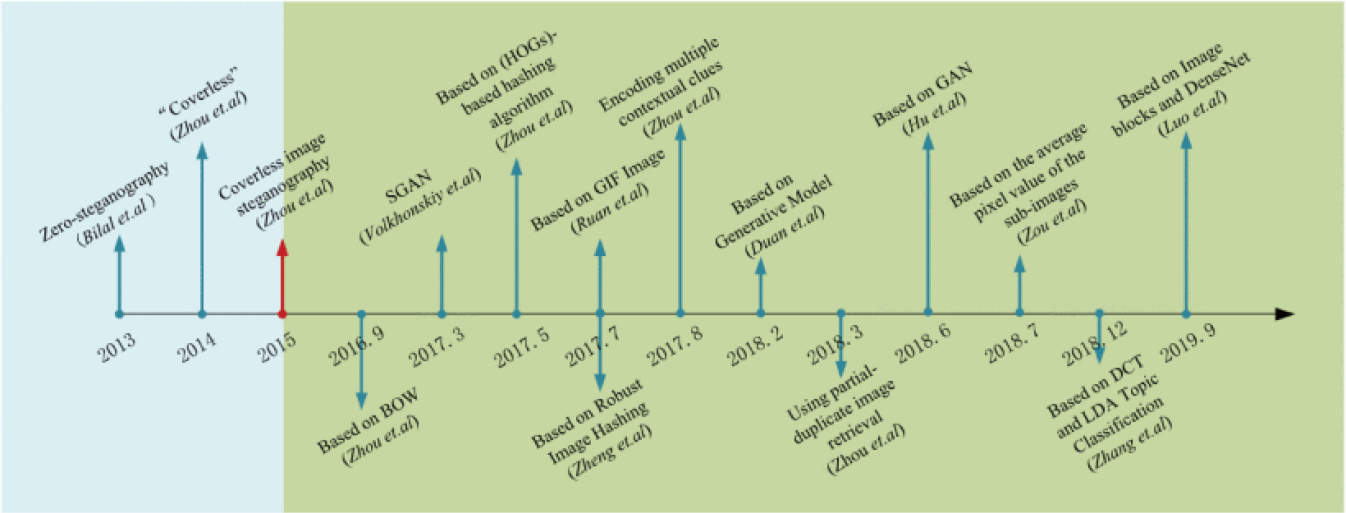


Figure 1: Summary of recent developments in coverless image steganography as described by Qin et al. [10] (please refer to the List of Acronyms).

value, colour, texture, edge, contour and high-level semantics. This means that the stego-image (i.e., the cover image containing the hidden data) is not altered, compared with the original cover. As a consequence, current steganalysis methods cannot detect the covert channel established by coverless image steganography, and recover the hidden data. The state-of-the-art of research in this field is summarised in Figure 1. From this evolution timeline, 11 methods were identified and compared, e.g., in terms of information hiding process, hidden data extraction process, and hidden data transmission process. From this preliminary analysis, the authors also identified the following recurring stages for all the coverless image steganography methods studied: (1) "Preprocessing", (2) "Feature extraction", (3) "Generation of hash sequence", and (4) "Mapping relationships". Stage (3) is used in order to resist cover image attacks (e.g., re-scaling and changes in contrast and luminosity); this means that the hash sequence of the cover image should not alter during transmission. Stage (4) is then required to map segments of hidden information and hash sequences. They further classified existing methods for coverless image steganography into 2 categories: methods which use the original attributes of the cover image to hide data, and methods which use a noise vector to train the generator model of a GAN to produce a cover image. The main disadvantage of the former is its capacity to hide data which is constrained by the length of the cover image hash value. The main disadvantage of the latter is the quality and integrity of the image generated since they cannot be guaranteed. Nevertheless, the latter is very robust to attacks since the only way to reveal the hidden information is to have access to the GAN model.

Editorial Notes

The survey by Qin et al. [10] has English writing issues, and its structure could be clearer. However, we believe that those factors do not diminish the technical value of the article.

Image Steganalysis

Yedroudj et al. [22] proposed a new way to enrich a database in order to improve the convolutional neural network (CNN) based steganalysis performance. CNN-based steganalysis approaches perform better for larger learning databases. However, working with a large database with controlled acquisition conditions is usually rare or unrealistic in an operational context. They first duplicated the learning set and added a particular noise for database augmentation, which is a classical principle in machine learning, especially in image classification. However, it did not give compelling results for their CNN-based steganalysis. They then used the "pixels-off" noise principle where they switched off a small proportion of pixels from the duplicated initial database of cover images. This approach is efficient, generic, and is usable in conjunction with other data-enrichment approaches. Additionally, it can be used to build an

© 2021 University of Kent, UK

informed database that they named "Side-Channel-Aware databases" (SCA-databases).

Editorial Notes

Yedroudj et al. [22] point out an interesting counter-intuitive behaviour specific to steganalysis using deep-learning. They postulated and verified that pixel-off noise increases the error probability for feature-based steganalysis.

You et al. [23] point out that CNN provides powerful capabilities for image steganalysis. However, there are still few reliable CNN-based methods for applying steganalysis to images of arbitrary size. This is mainly due to the particularity of steganographic signals. In this paper, the authors have addressed this issue by exploring the possibility of exploiting a network for steganalysing images of varying sizes without retraining its parameters. On the assumption that natural image noise is similar between different image sub-regions, they have proposed an end-to-end, deep learning, novel solution for distinguishing steganography images from normal images. Their solution provides satisfying performance. The proposed network first takes the image as the input, then identifies the relationships between the noise of different image sub-regions, and, finally, outputs the resulting classification based upon them. Their algorithm adopts a Siamese, CNN-based architecture, which consists of two symmetrical subnets with shared parameters, and contains three phases: pre-processing, feature extraction, and fusion/classification. To validate the network, they generated datasets composed of steganography images with multiple sizes and their corresponding normal images sourced from BOSSbase 1.01 and ALASKA #2. Experimental results produced by the data generated by various methods show that the proposed network is well-generalised and robust.

Editorial Notes

Other than being an interesting paper, the work by You et al. [23] has high tutorial value. They put effort in summarising three main image steganography approaches that nicely places their work in the appropriate context.

Tan et al. [14] point out that structure expansion (resulting in detection performance improvements of deep-learning based steganalysers) when done excessively, results in huge computational cost, storage overheads, and consequently difficulty in training and deployment. The authors proposed CALPA-NET, a ChAnneL-Pruning-Assisted deep residual network architecture search approach to shrink the network structure of existing vast, over-parameterised deep-learning based steganalysers. They observed that the broad inverted-pyramid structure of existing deep-learning based steganalysers might contradict the well-established model diversity oriented philosophy, and therefore is not suitable for steganalysis. They introduced a hybrid criterion combined with two network pruning schemes, to adaptively shrink every involved convolutional layer in a data-driven manner. The resulting network architecture presented a slender bottleneck-like structure. They conducted extensive experiments on BOSSBase + BOWS2 dataset, more diverse ALASKA dataset and even a large-scale subset extracted from ImageNet CLS-LOC dataset. The experimental results show that the model structure generated by their proposed CALPA-NET can achieve comparative performance with less than two percent of parameters and about one third FLOPs compared to the original steganalytic model. The new model possesses even better adaptivity, transferability, and scalability.

Editorial Notes

As has been mentioned in the work of Tan et al. [14], the horizon for the discipline of deep-learning based steganalysers is a completely automatic steganalytic framework generation.

© 2021 University of Kent, UK

Fake Software and Services

Introduction

We discuss some of the most recent papers on fake mobile apps and fake web-based services. Attackers use such systems to typically get private information about the victims. These works range from creating a repository of such fake apps, techniques used by the fake app developers to trap users, and the various statistical and other techniques that may be used to devise detection and screening mechanisms. Many of these works have published the software they have developed and their fake-app related repositories for further research.

Fake and Malicious Mobile Apps

Wapet et al. [18] took up the problem of detecting fake apps impersonating organisations in *mobile market places* (MMP). They pointed out that there are a significant number of such illegitimate apps, resulting in negative publicity for MMPs. All previous scanning solutions in this domain only focus on the detection of illegitimate apps which mimic existing ones. However, a new category of attacks emerged, where fake apps for enterprises, who are yet to publish their app on the MMP, have appeared. Thereby, an attacker may be one step ahead and publish a malicious app using the graphic identity of a trusted enterprise. The common previous solutions like Androguard and FsQuadra could not address this issue. Famous enterprises such as Blackberry, Netflix, and Niantic (Pokemon Go) have been victims of such attacks. In this work, the authors designed and implemented a security check system called IMAD (Illegitimate Mobile App Detector) which is able to limit the aforementioned attacks. The evaluation results show that IMAD can protect companies from such attacks with an acceptable error rate and at a low cost for MMPs. Figure 2 shows the synthetic app submission workflow, from the security checkpoint point of view, where IMAD can be plugged in.

Tang et al. [16] claimed to have conducted the first systematic and comprehensive empirical study on a large set of fake apps. These were without official certificates but simulated the corresponding official apps, or looked almost identical to their official corresponding apps. The ultimate goal was to evaluate downloads or malicious behaviours. In this paper, they presented discoveries from three different perspectives, namely fake sample characteristics, a quantitative study on fake samples and development trends. Moreover, valuable domain knowledge, like fake apps' naming tendency and fake developers' evasive strategies, were then presented. They obtained a representative ranked list of apps from the online big data analysis service provider Analysys (`https://www.analysys.cn/`). They collaborated with their partner company Pwnzen Infotech Inc. (`http://pwnzen.com/`), one of the leading security companies in China, to collect the app samples. Among them, 52,638 fake samples were identified for further analysis. In order to work with the large-scale data, they used the common industrial practice of analysing subjects' metadata. They identified and extracted 8 metadata items to support their comprehensive measurement. The list of apps targeted for analysis and their related statistics are shown in Figure 3.

Editorial Notes

As in Tang et al. [16], partnerships with key players from the security industry are essential for these ever-evolving domains to take advantage of their formidable expertise and state-of-the-art infrastructure.

Shi et al. [12] pointed out that the most common way to spread Android malware used to be through repackaging popular benign apps with malicious payload. An alarming new trend in the Android ecosystem has been observed since 2016: a growing number of Android malware samples abuse recent app-virtualisation innovation as a new distribution channel. App-virtualisation enables a user to run multiple copies of the same app on a single device. This convenience is now being availed by tens of millions of users. An app-virtualisation platform allows the flexibility to launch arbitrary plugins without the hassle of installation. This allows cybercriminals to repackage various malicious Android Application Package (APK) files as plugins into the app-virtualisation platforms. They bypass anti-malware scanners by hiding the grafted malicious payload in plugins, and it also defies the basic premise embodied by existing repackaged app detection solutions.

© 2021 University of Kent, UK

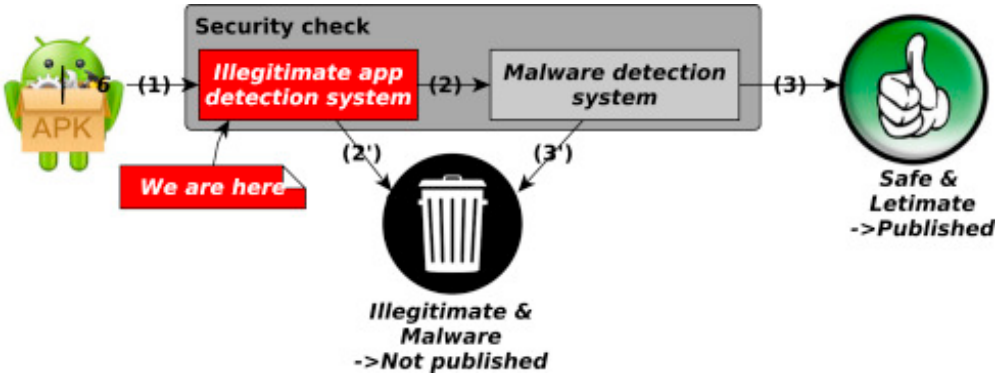


Figure 2: The synthetic app submission workflow, from the security checkpoint point of view – as pointed out by Wapet et al. [18].

As app-virtualisation-based apps are not necessarily malware, in this paper, the authors aimed to make a verdict on them *prior to run time*. This in-depth study resulted in two key observations: 1) the proxy layer between plugin apps and the Android framework is the core of app-virtualisation mechanism, and it reveals the feature of finite state transitions; 2) malware typically loads plugins stealthily and hides malicious behaviour. These insights led them to develop a two-layer detection approach, called VAHunt. They first designed a stateful detection model to identify the existence of an app-virtualisation engine in APK files. Then, they performed data flow analysis to extract fingerprinting features to differentiate between malicious and benign loading strategies. VAHunt was tested for a considerable period of time in Antiy AVL Mobile Security, a leading mobile security company, to detect more than 139,000 app-virtualisation-based samples. VAHunt achieved 0.7% false negatives and zero false positive. Their automated detection frees security analysts from the burden of reverse engineering. Figure 4 shows some popular app-virtualisation apps and engines.

Editorial Notes

Portability often adds vulnerability. As in Shi et al. [12], detaching a system from the underlying layer takes away the control of that layer. Every aspect of that control is lost - including the protection provided by that layer.

Xia et al. [20] claimed to be the first to identify and characterise cryptocurrency exchange scams. As hundreds of cryptocurrency exchanges are emerging to facilitate the trading of digital assets, they are attracting the attention of attackers. A number of scam attacks were reported targeting cryptocurrency exchanges, leading to a huge amount of financial loss. The authors attempted to identify such scams by first identifying more than 1500 scam domains and over 300 fake mobile apps, by collecting existing reports and using typosquatting (mistyped URL) generation techniques. They investigated the relationship between the scam domains and fake apps. They identified 94 scam domain families and 30 fake app families. By further characterising the impacts of such scams, it was revealed that these scams have incurred financial loss of at least 520,000 US dollars. They further observed that the fake apps were sneaked to major app markets (including Google Play) to infect unsuspicious users. The findings in this paper demonstrated the urgency to identify and prevent cryptocurrency exchange scams.

Editorial Notes

Xia et al. [20] have publicly released (at `https://cryptoexchangescam.github.io/ScamDataset/`) all the identified scam domains and fake apps, to facilitate future research. It will be interesting to have a comprehensive study of all the attack techniques used.

Hu et al. [3] claimed to have presented the first in-depth measurement study of *app squatting* showing its prevalence and implications. Domain squatting, the adversarial tactic where attackers register domain names that mimic popular ones, has been observed for decades. However, there has been growing anecdotal evidence that this style of attack has

© 2021 University of Kent, UK

Name	Category	MAI (Monthly Activeness Indicator)	Update Frequency (day/version)	#Total	#Fake	Fake Sample Rate	Avg Fake Latency (day)
WeChat*	**SocialNetwork**	**91.2K**	**6.4**	**9248**	**6447**	**69.7%**	**12.1**
QQ*	**SocialNetwork**	**54.6K**	**10.7**	**11167**	**3780**	**33.8%**	**9.2**
iQiyi	Video	53.5K	6.4	7586	3481	45.9%	9.3
Alipay	Life	48.1K	10.2	983	231	23.5%	10.1
Taobao*	**OnlineShopping**	**47.5K**	**7.0**	**6003**	**3010**	**50.1%**	**8.1**
TencentVideo	Video	47.3K	6.3	1429	68	4.8%	10.7
Youku	Video	40.9K	7.3	2058	262	12.7%	6.7
Weibo*	**SocialNetwork**	**39.2K**	**5.3**	**5947**	**2715**	**45.7%**	**5.7**
WiFiMasterKey	SystemTool	36.4K	3.1	4808	2999	62.4%	3.0
SougouInput	SystemTool	33.3K	11.0	898	40	4.5%	21.8
MobileBaidu	Information	32.4K	11.1	15651	3514	22.5%	12.8
TencentNews	Information	28.7K	8.5	1051	11	1.0%	8.9
QQBrowser	Information	27.8K	5.6	1369	43	3.1%	11.6
Toutiao	Information	27.4K	4.4	3538	179	5.1%	5.6
Myapp	AppStore	27K	11.4	2419	266	11.0%	11.6
Kwai	Video	24.4K	3.2	8273	4270	51.6%	3.5
WeSecure	SystemTool	24.2K	8.7	2463	1340	54.4%	8.7
Amap	Life	24K	6.5	1225	51	4.2%	13.1
KugouMusic	Music	23K	8.6	1313	122	9.3%	12.2
QQMusic	Music	21.7K	9.4	1132	65	5.7%	14.6
BaiduMap	Life	21.3K	8.8	2609	1438	55.1%	15.3
TikTok	Video	19.4K	11.1	317	12	3.8%	8.3
JD*	**OnlineShopping**	**18.5K**	**10.9**	**5000**	**2377**	**47.5%**	**12.3**
UCBrowser	Information	16.7K	7.4	4232	1624	38.4%	7.0
360Security	SystemTool	15.4K	12.4	3670	1423	38.8%	19.1
TencentKaraoke	Music	14.7K	21.1	618	215	34.8%	17.3
MeiTuan	Life	13K	8.0	4752	1415	29.8%	6.9
Pinduoduo*	**OnlineShopping**	**12.9K**	**6.6**	**2327**	**551**	**23.7%**	**7.8**
ArenaofValor*	**Game**	**12.5K**	**15.5**	**2350**	**1319**	**56.1%**	**12.3**
MeiTuXiuXiu	Camera	12.4K	5.4	1705	784	46.0%	5.8
VigoVideo	Video	12.2K	11.9	410	16	3.9%	9.6
MojiWeather	Life	12K	4.2	10081	7093	70.4%	4.7
DiDi	Life	11.8K	8.6	943	117	12.4%	7.0
HuaweiAppStore	AppStore	11.8K	N/A	0	0	0.0%	N/A
HappyElements*	**Game**	**11.2K**	**19.7**	**2406**	**1738**	**72.2%**	**20.6**
KuwoMusicPlayer	Music	11K	2.9	3778	69	1.8%	4.2
iXigua	Video	11K	11.5	866	100	11.5%	8.8
OPPOAppStore	AppStore	10.8K	N/A	0	0	0.0%	N/A
CleanMaster	SystemTool	9.9K	10.3	1803	388	21.5%	13.5
360CleanDroid	SystemTool	9.6K	17.3	327	8	2.4%	8.5
360Zhushou	AppStore	9.2K	7.6	1616	137	8.5%	8.4
TencentWiFiManager	SystemTool	8.8K	19.5	1636	658	40.2%	15.7
XunfeiInput	SystemTool	8.6K	6.0	1451	8	0.6%	10.1
BaiduAppSearch	AppStore	8.2K	11.4	3849	437	11.4%	14.5
MiAppStore	AppStore	7.8K	N/A	0	0	0.0%	N/A
WPSOffice*	**Productivity**	**7.4K**	**6.0**	**1152**	**69**	**6.0%**	**7.8**
BeautyCam	Camera	7.1K	5.3	1600	691	43.2%	6.3
NeteaseCloudMusic	Music	7K	10.5	616	6	1.0%	12.2
NeteaseNews	Information	6.7K	7.0	1441	93	6.5%	5.0
QQMail*	**Productivity**	**6.6K**	**16.4**	**520**	**11**	**2.1%**	**10.4**

* Detailed descriptions are given in **Answer to RQ 2.4**

Figure 3: List of the apps targeted for analysis by Tang et al. [16] and their related statistics.

spread to other areas. In this paper, the authors explored the presence of squatting attacks in the mobile app ecosystem. In "App Squatting", attackers release apps with identifiers (e.g., app name or package name) that are confusingly similar to those of popular apps or well-known Internet brands. The authors first identified 11 common deformation approaches of app squatters and propose "AppCrazy", a tool for automatically generating variations of app identifiers. They applied AppCrazy to the top 500 most popular apps in Google Play, generating 224,322 deformation keywords. These keywords were then used to test for app squatters on popular markets. This confirmed the scale of the problem, identifying 10,553 squatting apps - *an average of over 20 squatting apps for each legitimate one*. Their investigation revealed that more than 51% of the squatting apps were malicious, with some being extremely popular (up to 10 million downloads). They also found that mobile app markets have not been successful in identifying and eliminating squatting apps. Their

© 2021 University of Kent, UK

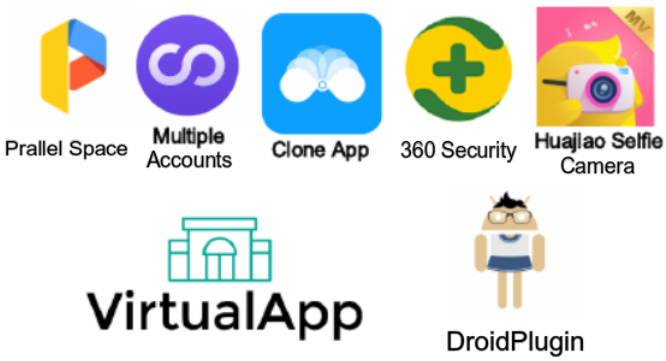


Figure 4: Popular app-virtualisation apps and engines pointed out by Shi et al. [12].

findings demonstrated the urgency to identify and prevent app squatting abuses.

Editorial Notes

Hu et al. [3] have publicly released all the identified squatting apps, as well as their tool AppCrazy (`https://github.com/squattingapp/AppCrazy`). It will be interesting to know the vulnerability of mobile phone users to these scams, across demographics.

Li et al. [5] conducted a systematic literature review of repackaging mobile apps which is a serious threat to the Android ecosystem. It is used by plagiarists who clone apps from other developers, say for redirecting advertisement revenue and thus deprives app developers of their benefits. It is also used by malware writers who piggyback malicious payloads on popular apps to spread malware, and increases the workload of market maintainers. In the space of six years, the research around this specific issue has produced 57 approaches which do not readily scale to millions of apps or are only evaluated on private datasets without, in general, tool support available to the community. They have highlighted the shortcomings of these approaches in terms of the impracticality of the approaches, lack of reproducibility, and sub-optimal evaluation scenarios. They have provided a large dataset that supports replications of existing solutions and implications of new research directions. Their work builds upon the popular AndroZoo repository, which can serve as an exchange repository for describing a dataset using the hash values of apps. They also enumerated research directions that the community should take up for advancing the state-of-the-art in the topic.

Editorial Notes

Uniquely identifying a digital object using hash functions is a very well known technique. It gets easily translated into equality testing of two objects. An easy way to evade such techniques would be to introduce minor changes in the object so that it still works as it used to, but has a different hash value.

Wu et al. [19] pointed out that with low cost and high profit, fake IoT apps are an increasing risk to the security of the IoT ecosystem. They proposed a novel fake IoT app detection method, referred as MSimDroid, based on multidimensional similarity to mitigate the threat. MSimDroid focuses on the distribution channels of fake apps (app markets). It consists of whole app similarity, resource similarity, code similarity, and their joint strategy. For similarity calculation, they designed a distinctive algorithm based on the feature of different fake patterns. For joint strategy, which is the scheduler of multiple algorithms, the accuracy and time consumption of MSimDroid was balanced. Experiments demonstrated that the accuracy of MSimDroid is more than 99.31% on ground-truth data set and 97.43% in the wild. The IoT apps from multiple well-known app markets revealed that the average proportion of fake apps is about 14.66%, and that of mixed-mode apps (including IoT and non-IoT apps) is 10.78%. Besides, it found that about 0.58% of IoT apps suffer from malice, while the average ratio of mixed-mode apps is 1.06%.

Fake Antivirus Software

Koide et al. [4] proposed a system to automati-

© 2021 University of Kent, UK

cally crawl the web and identify *fake removal information advertisement* (FRAD) sites. Fake antivirus (AV) software is a serious threat on the Internet to make users install malware and expose their personal information. FRAD sites, which introduce fake removal information for cyber threats, have emerged as platforms for distributing fake AV software. Although FRAD sites seriously threaten users who have been suffering from cyber threats and need information for removing them, little attention has been given to revealing these sites. The authors performed a comprehensive analysis of both passively and actively collected data to demonstrate the pervasiveness of this type of attack. Their system collected 2,913 FRAD sites in 31 languages, which have 73.5 million visits per month in total. They showed that FRAD sites occupy search results when users search for cyber threats, thus preventing the users from obtaining the correct information.

© 2021 University of Kent, UK

Data Poisoning in AI Systems

Introduction

Data poisoning is a technique which aims to inject *poisonous data* into the training set in order to make an AI model misbehave. Consider modern industrial scale applications of machine learning systems, where data collection and policy updates are done in a distributed way. In such applications, it is easy for an attacker to have access to the learner's training data, and the power to manipulate a fraction of that data in order to make the learner satisfy certain objectives. Recent research shows that data poisoning is a useful way of performing adversarial attacks, including backdoor/trojan attacks in which an AI model is manipulated to misbehave only for a particular input.

Poisoning Attacks

Liu and Shroff [8] investigated data poisoning attacks against stochastic multi-armed bandit algorithms which form a class of online learning problems with limited feedback. These online learning problems have important applications in online recommendation systems and adaptive medical treatment. The authors proposed an optimisation-based framework for offline data poisoning attacks and three algorithm-specific offline attack strategies against ϵ-greedy, Upper Confidence Bound (UCB) and Thompson Sampling which are common ways to solve multi-armed bandits. Besides this, they introduced an adaptive attack strategy that can hijack any bandit algorithm without knowing it for online data poisoning attacks. They evaluated their attack strategies using theoretical results. Bandit algorithms are widely employed in real-world applications, and these results expose a significant security threat.

Editorial Notes

The attack strategy proposed by Liu and Shroff [8] is an adaptive attack strategy that "*can hijack any bandit algorithm without knowing the bandit algorithm*". According to the authors, this is the first negative result showing that there is no robust and good stochastic bandit algorithm that can survive online poisoning attack.

Zhu et al. [25] presented a transferable clean-label poisoning attack for Deep Neural Networks (DNNs), namely *Convex Polytope Attack* (CP), which aims to inject correctly labelled poisonous images into training data to misclassify a targeted image. The idea behind the proposed attack is to construct a convex polytope around the target image in feature space, so that a linear classifier which overfits the poisoned dataset is guaranteed to classify the target into the poisoned class. The authors evaluated the proposed attack on the CIFAR-10 image classification dataset together with eight different architectures and compared the results with the results of another clean-label poisoning attack, *Feature Collision Attack* (FC). The results showed that FC never achieved a success rate higher than 50%, while CP achieved success rates higher or close to 50% in most cases. Figure 5 shows a qualitative example on the comparison between FC and CP attacks.

Editorial Notes

Zhu et al. [25] have released their code for the experiments at `https://github.com/zhuchen03/ConvexPolytopePosioning`.

Liu et al. [9] proposed a backdoor scanning technique for neural network based AI models, called Artificial Brain Stimulation (ABS), to detect trojaned models. Potentially compromised neurons substantially elevate the activation of a specific output label regardless of the input. The proposed technique includes analysing inner neuron behaviours to identify such neurons. To confirm that a neuron is truly compromised, the proposed technique suggests reverse engineering the trojan trigger through an optimisation procedure. This entails identifying some specific pattern that the input is stamped with. Figure 6 shows some example triggers and their reverse engineered versions. The authors evaluated the proposed technique on 177 trojaned models and 144 benign models. These models belong to seven different model structures and six different datasets. The results showed that the proposed system achieved over 90% detection rate for most cases. In addition, the authors compared the results with the evaluation results of Neural Cleanse (NC) which is another state-of-the-art backdoor scanning technique.

© 2021 University of Kent, UK

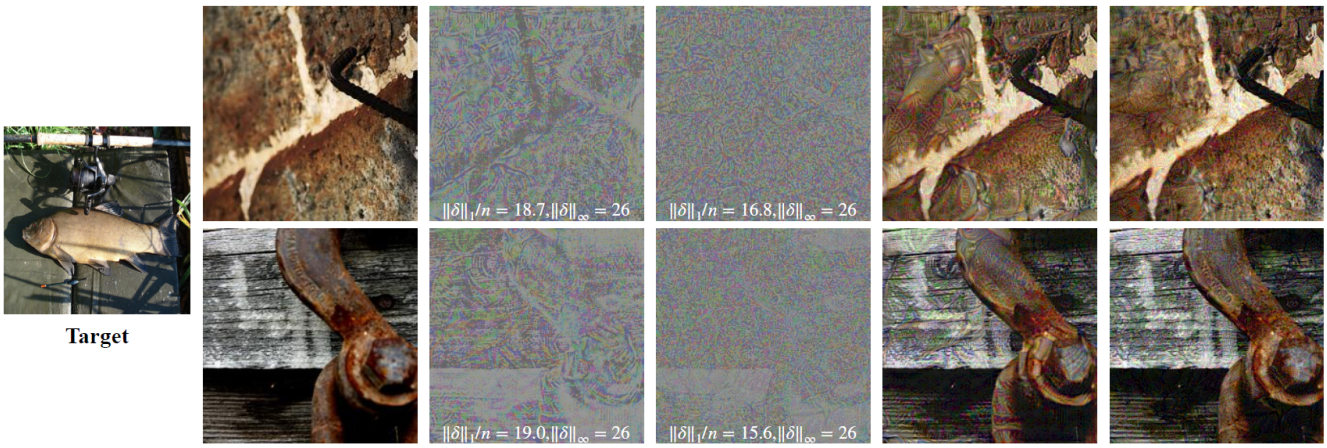


Figure 5: A qualitative example of the difference in poison images generated by Feature Collision (FC) Attack and Convex Polytope (CP) Attack given by Zhu et al. [25]. Both attacks aim to make the model misclassify the target fish image on the left into a hook.

Editorial Notes

Liu et al. [9] used the technique of invariance of a model's behaviour to vary inputs for detecting backdoors. This is an interesting direction for categorising and exploring backdoors in the broader context of security research.

Backdoor Attacks

Lin et al. [6] introduced *composite backdoor attack* for DNNs, which leverages existing benign features of multiple output labels to compose a trojan trigger rather than injecting a patch by data poisoning. As exemplified in Figure 7, the proposed attack causes misclassification when a combination of selected labels is present in a sample. The authors developed an attack engine which trains the trojaned model from scratch or retrains a pre-trained model to inject the backdoor by following a data poisoning-based trojan training procedure. This procedure involves an existing training set and a mixer which decides how to combine benign features. The mixer is then used to synthesise new training samples to combine features from the trigger labels. The authors stated that composite attack is harder to be detected than patch-based attacks since the proposed attack avoids establishing strong correlations between a few neurons and the target label by reusing the existing features. To evaluate the proposed attack, the authors injected backdoors in seven tasks, including object recognition, traffic sign recognition, face recognition, topic classification, and three different object detection tasks. The results showed that composite attacks achieved more than 80% attack success rate while the trojaned model preserved the same level of accuracy for each of the tasks. The authors also tried to detect the proposed attack by using two state-of-the-art backdoor scanners, Neural Cleanse and ABS [9], and they found that none of them could detect the injected backdoors. Lastly, the study was concluded with a possible defence approach against the composite attack in spite of the limitations noted by the authors.

Editorial Notes

Lin et al. [6] proposed a novel technique, *composite backdoor attack*, which seems to be quite impressive considering that they could evade state-of-the-art backdoor scanners. The proposed attack uses benign features which makes it harder to be detected. Although a possible defence strategy was given in the paper, the authors reported that it was very preliminary and had several limitations.

Tan and Shokri [15] designed an adversarial backdoor embedding algorithm for deep learning, which can bypass several existing detection algorithms. More precisely, the authors proposed an adaptive

© 2021 University of Kent, UK

(a) Triggers

(b) Reverse Engineered Triggers

Figure 6: Triggers and Reverse Engineered Triggers presented by Liu et al. [9].

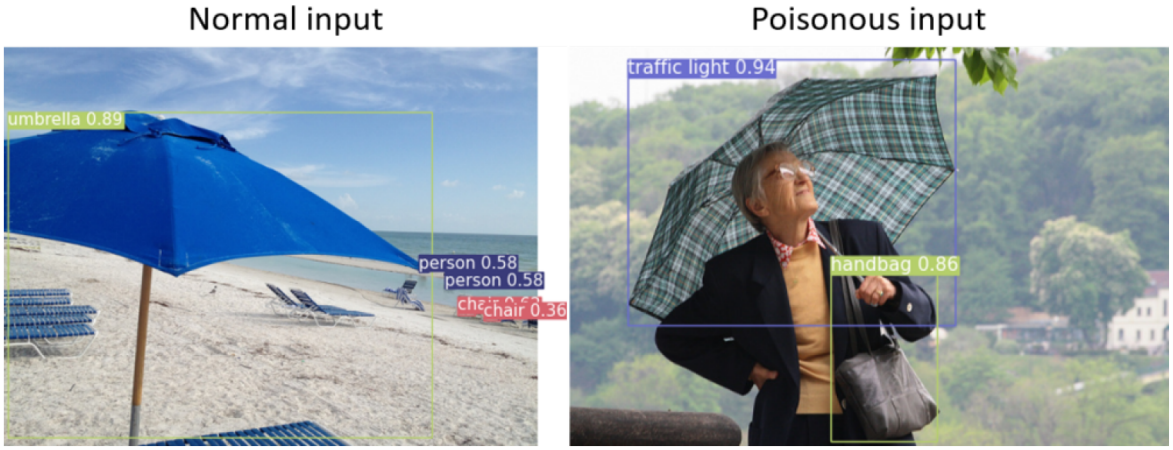


Figure 7: Example of composite attack on object detection given by Lin et al. [6]. Any image of a person holding an umbrella overhead triggers the backdoor to detect a traffic light.

adversarial training algorithm that maximises the original loss function of the model, and the latent indistinguishability between adversarial inputs and benign inputs. The authors evaluated the proposed algorithm on different defence approaches, including dataset filtering using spectral signatures, dataset filtering using activation clustering and feature pruning, and used two image classification datasets, CIFAR-10 and German Traffic Sign Recognition Benchmark (GTSRB), together with two model architectures, DenseNet-BC and VGG. The results showed that the proposed algorithm could bypass each of the aforementioned detection algorithms.

© 2021 University of Kent, UK

Vulnerabilities in AI Systems

Introduction

Recent research on AI/ML security has shown that AI models are vulnerable to several types of attacks. In that vein, this section includes recent security analyses of AI systems and recently proposed attacks exploiting the identified vulnerabilities of different AI models.

Security Analyses of AI Systems

Zhang et al. [24] conducted a systematic study on the security of interpretable deep learning systems (IDLSes). Firstly, the authors drew attention to the importance of interpretability of DNNs which can help to understand the inner workings of DNNs and identifying adversarial manipulations. In this manner, the authors presented ADV^2, a new class of attacks that generate adversarial inputs not only misleading target DNNs but also deceiving their coupled interpretation models. To evaluate the proposed attack, the authors conducted an empirical study on DNNs, ResNet-50 and DenseNet-169, and interpreters, GRAD, CAM, RTS and MASK, by using the ImageNet dataset. The results showed that ADV^2 attack reached above 95% success rate for all cases. The authors also showed that the proposed attack can generate adversarial inputs with interpretations highly similar to benign cases, both qualitatively and quantitatively (with L_1 and L_2 measures and intersection-over-union (IoU) test). In addition, they identified that DNNs and their interpretation models are often misaligned, which makes it possible to exploit both models simultaneously. Finally, the authors explored potential countermeasures against ADV^2, including leveraging its low transferability and incorporating it in an adversarial training framework.

Ling et al. [7] presented the design, implementation, and evaluation of DEEPSEC, a uniform platform that enables researchers and practitioners to measure the vulnerability of Deep Learning (DL) models, evaluate the effectiveness of various attacks/defences, and conduct comparative studies on attacks/defences. DEEPSEC incorporates 16 state-of-the-art attacks with 10 attack utility metrics, and 13 state-of-the-art defences with 5 defensive utility metrics. Figure 8 shows the overview of the proposed platform. The authors used the proposed platform to evaluate the attacks and defences implemented in DEEPSEC in terms of misclassification, imperceptibility, robustness and computation cost on MNIST and CIFAR-10 datasets. Lastly, they demonstrated the functionality of the proposed platform with two case studies.

Editorial Notes

DEEPSEC implemented by Ling et al. [7] have provided various attacks and defences together with utility metrics for their evaluation. Even though the platform only focuses on non-adaptive and white-box attacks, it enables easy execution of comparative studies. The authors have shared their code at `https://github.com/kleincup/DEEPSEC`

Attacks Exploiting Vulnerabilities in AI Systems

Sun et al. [13] introduced two attacks against Deep Reinforcement Learning (DRL) agents for an adversary to inject adversarial samples in a minimal set of critical moments while causing the most severe damage to the agent. The first attack named Critical Point Attack, aims to discover the fewest critical moments to achieve the most severe damage to the agent. The second attack, named Antagonist Attack, aims to automatically discover the optimal attack strategy using the lowest attack cost without any domain knowledge. The authors used different DRL applications, including Atari games (Pong and Breakout), autonomous driving (TORCS - The Open Racing Car Simulator) and continuous robot control (Mojuco) as benchmarks for the evaluation of the proposed attacks. The results showed that the proposed attacks were successful in compromising the above-mentioned DRL tasks. In addition, the authors reported that the proposed attacks are generic, and they require fewer time steps to perform a successful attack compared to other attacks proposed in the literature.

Salem et al. [11] addressed membership inference attacks by which information can be extracted from the training set in Machine Learning as a Service (MLaaS) applications. The authors aimed to

© 2021 University of Kent, UK

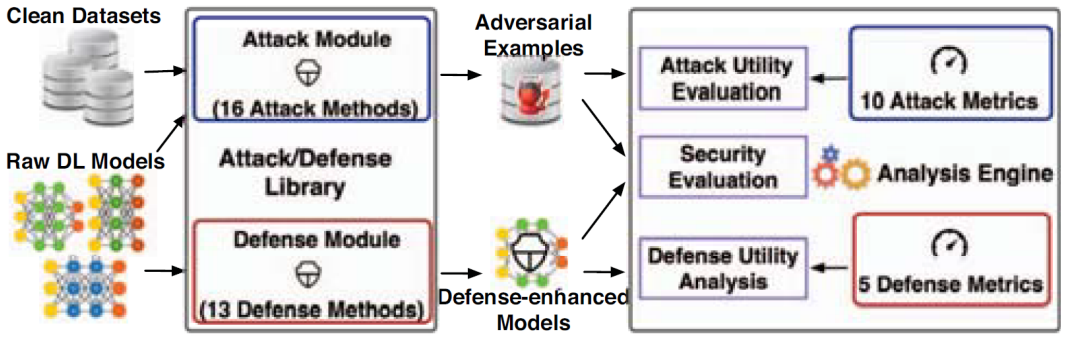


Figure 8: The System Overview of DEEPSEC proposed by Ling et al. [7].

	Multi-image Attack		Singe-image Attack	
Scenario	Attack Success Rate	Model Accuracy	Avg Attack Success Rate	Avg Model Accuracy
Traffic Sign	100%	88.8%	67.1%	87.4%
Iris Identification	90.8%	96.2%	77.1%	97.7%
Politician Recognition	99.8%	97.1%	90.0%	96.7%

Figure 9: Attack performance of latent backdoors in real-world scenarios, proposed by Yao et al. [21].

show that such attacks are broadly applicable at low cost even if several key assumptions on the adversary were relaxed – including using multiple so-called shadow models, knowledge of the target model structure, and having a dataset from the same distribution as the target model's training data. For each key assumption, the authors evaluated their claims under different adversarial setups on, in total, eight different datasets. Besides this, they used Google's MLaaS, Google Cloud Prediction API, to evaluate their attacks in a real-world setting. The results showed that relaxing above-mentioned key assumptions still provided similar results for attack success rates, meaning that membership inference attacks can be more severe than it has been thought to be. Lastly, the authors proposed two defence techniques for the attacks they performed, namely Dropout and Model Stacking. These techniques aimed to increase the generalisability of ML models to avoid overfitting, which was reported by the authors to be the reason of the successful membership inference attacks.

Yao et al. [21] proposed *latent backdoors*, a variant of backdoor attacks that can function under transfer learning, which is used to reduce the vulnerability of DNN models to backdoor attacks by customising pretrained *Teacher* models rather than training models from scratch. The authors described latent backdoors as incomplete backdoors embedded into a *Teacher* model, and automatically inherited by multiple *Student* models through transfer learning. The authors evaluated the proposed attack by considering four classification tasks, including handwritten digit recognition, traffic sign recognition, face recognition and iris identification, and they used datasets suitable for each task such as MNIST, GTSRB, Laboratory for Intelligent & Safe Automobiles (LISA), VGG-Face and PubFig. The results showed that the proposed attack achieved over 96% attack success rate for all tasks without compromising the model accuracies when the attacker was able to obtain multiple target images. The authors also reported that the proposed attack still produced high attack success rates when the attacker had only a single target image, but the dataset was large enough. Lastly, the authors evaluated latent backdoors on three real-world classification scenarios, which are traffic sign recognition, iris-based user identification and facial recognition of politicians, and they managed to obtain similar results, as shown in Figure 9.

© 2021 University of Kent, UK

References

[1] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. 2010. Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing* 90 (2010), 727–752. Issue 3. https://doi.org/10.1016/j.sigpro.2009.08.010

[2] Rémi Cogranne, Quentin Giboulot, and Patrick Bas. 2020. Steganography by Minimizing Statistical Detectability: The Cases of JPEG and Color Images. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*. ACM, 161–167. https://doi.org/10.1145/3369412.3395075

[3] Yangyu Hu, Haoyu Wang, Ren He, Li Li, Gareth Tyson, Ignacio Castro, Yao Guo, Lei Wu, and Guoai Xu. 2020. Mobile App Squatting. In *Proceedings of The Web Conference 2020*. ACM, 1727–1738. https://doi.org/10.1145/3366423.3380243

[4] Takashi Koide, Daiki Chiba, Mitsuaki Akiyama, Katsunari Yoshioka, and Tsutomu Matsumoto. 2020. It Never Rains but It Pours: Analyzing and Detecting Fake Removal Information Advertisement Sites. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 171–191. https://doi.org/10.1007/978-3-030-52683-2_9

[5] Li Li, Tegawende F. Bissyande, and Jacques Klein. 2019. Rebooting Research on Detecting Repackaged Android Apps: Literature Review and Benchmark. *IEEE Transactions on Software Engineering* (2019), 1–18. https://doi.org/10.1109/TSE.2019.2901679 (In press).

[6] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite Backdoor Attack for Deep Neural Network by Mixing Existing Benign Features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 113–131. https://doi.org/10.1145/3372297.3423362

[7] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. 2019. DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model. In *2019 IEEE Symposium on Security and Privacy*. IEEE, 673–690. https://doi.org/10.1109/SP.2019.00023

[8] Fang Liu and Ness Shroff. 2019. Data Poisoning Attacks on Stochastic Bandits. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 4042–4050. http://proceedings.mlr.press/v97/liu19e.html

[9] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. ABS: Scanning Neural Networks for Back-Doors by Artificial Brain Stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1265–1282. https://doi.org/10.1145/3319535.3363216

[10] Jiaohua Qin, Yuanjing Luo, Xuyu Xiang, Yun Tan, and Huajun Huang. 2019. Coverless Image Steganography: A Survey. *IEEE Access* 7 (2019), 171372–171394. https://doi.org/10.1109/ACCESS.2019.2955452

[11] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Proceedings of 2019 Conference on Network and Distributed Systems Security Symposium*. 15. https://doi.org/10.14722/ndss.2019.23119

[12] Luman Shi, Jiang Ming, Jianming Fu, Guojun Peng, Dongpeng Xu, Kun Gao, and Xuanchen Pan. 2020. VAHunt: Warding Off New Repackaged Android Malware in App-Virtualization's Clothing. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 535–549. https://doi.org/10.1145/3372297.3423341

© 2021 University of Kent, UK

[13] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. 2020. Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 4 (2020), 5883–5891. https://doi.org/10.1609/aaai.v34i04.6047

[14] Shunquan Tan, Weilong Wu, Zilong Shao, Qiushi Li, Bin Li, and Jiwu Huang. 2021. CALPA-NET: Channel-Pruning-Assisted Deep Residual Network for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security* 16 (2021), 131–146. https://doi.org/10.1109/TIFS.2020.3005304

[15] Te Juin Lester Tan and Reza Shokri. 2020. Bypassing Backdoor Detection Algorithms in Deep Learning. In *Proceedings of 2020 IEEE European Symposium on Security and Privacy*. IEEE, 175–183. https://doi.org/10.1109/EuroSP48549.2020.00019

[16] Chongbin Tang, Sen Chen, Lingling Fan, Lihua Xu, Yang Liu, Zhushou Tang, and Liang Dou. 2019. A Large-Scale Empirical Study on Industrial Fake Apps. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*. IEEE, 183–192. https://doi.org/10.1109/ICSE-SEIP.2019.00028

[17] Yaofei Wang, Weiming Zhang, Weixiang Li, and Nenghai Yu. 2021. Non-Additive Cost Functions for JPEG Steganography Based on Block Boundary Maintenance. *IEEE Transactions on Information Forensics and Security* 16 (2021), 1117–1130. https://doi.org/10.1109/TIFS.2020.3029908

[18] Lavoisier Wapet, Alain Tchana, Giang Son Tran, and Daniel Hagimont. 2019. Preventing the Propagation of a New Kind of Illegitimate Apps. *Future Generation Computer Systems* 94 (2019), 368–380. https://doi.org/10.1016/j.future.2018.11.051

[19] Peng Wu, Dong Liu, Junfeng Wang, Baoguo Yuan, and Wenyuan Kuang. 2020. Detection of Fake IoT App Based on Multidimensional Similarity. *IEEE Internet of Things Journal* 7, 8 (2020), 7021–7031. https://doi.org/10.1109/JIOT.2020.2981693

[20] Pengcheng Xia, Haoyu Wang, Bowen Zhang, Ru Ji, Bingyu Gao, Lei Wu, Xiapu Luo, and Guoai Xu. 2020. Characterizing Cryptocurrency Exchange Scams. *Computers & Security* 98 (2020), 1–17. https://doi.org/10.1016/j.cose.2020.101993

[21] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2019. Latent Backdoor Attacks on Deep Neural Networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2041–2055. https://doi.org/10.1145/3319535.3354209

[22] Mehdi Yedroudj, Marc Chaumont, Frederic Comby, Ahmed Oulad Amara, and Patrick Bas. 2020. Pixels-off: Data-Augmentation Complementary Solution for Deep-Learning Steganalysis. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*. ACM, 39–48. https://doi.org/10.1145/3369412.3395061

[23] Weike You, Hong Zhang, and Xianfeng Zhao. 2021. A Siamese CNN for Image Steganalysis. *IEEE Transactions on Information Forensics and Security* 16 (2021), 291–306. https://doi.org/10.1109/TIFS.2020.3013204

[24] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. 2020. Interpretable Deep Learning Under Fire. In *Proceedings of 29th USENIX Security Symposium*. USENIX Association, 1659–1676. https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-xinyang

© 2021 University of Kent, UK

[25] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 7614–7623. `http://proceedings.mlr.press/v97/zhu19a.html`

© 2021 University of Kent, UK