ADVERSARIAL MACHINE LEARNING USING CONVOLUTIONAL NEURAL NETWORK WITH IMAGENET

Utsab Khakurel
Danda B. Rawat

Department of Electrical Engineering and Computer Science
Howard University
Washington DC 20059, USA
utsab.khakurel@bison.howard.com, danda.rawat@howard.com

ABSTRACT

Adversarial attacks are types of attacks where adversaries try to deceive the machine learning algorithm by providing deceptive input. Adversarial attacks are focused on providing inaccurate data at its training phase, or introducing maliciously designed data to deceive an already trained model. Adversarial instances are an excellent element of security to focus on because they represent a concrete challenge that AI is currently facing, and they are also a challenging component of security to work on because it requires a significant amount of research effort and time. Systems that are currently in use are very vulnerable to adversarial attacks. It is as simple as vandalizing traffic signs for the self-driving car to make mistakes. Any machine learning model is easy to attack, as we can feed them with malicious and wrong data input with ease. This paper focuses on the Adversarial manipulation of the Machine Learning Algorithm and illustrates how attacks are curated towards an Image-based trained model. A series of experiments based on the ImageNet images and pre-trained MobileNet, Convolutional Neural Network(CNN) model from TensorFlow is used to show how an adversarial attack based on Images can be curated to outwit the machine learning model. This paper will primarily focus on the AI/ML algorithm manipulation with one-pixel, multi-pixel, and all-pixel attacks on the TensorFlow pre-trained model.

Keywords: Machine learning, adversarial, one-pixel attack, multi-pixel attack.

1 INTRODUCTION

Artificial Intelligence(AI) and Machine Learning(ML) is an integral part of our modern life. Technology has brought an unforeseeable changes in human lives. Mobile phones, Computer, and IoT devices have taken over the world. They are already smarter than human in many levels. AI/ML is providing the capacity for these devices to learn and decide on their own. The growing volumes and varieties of data, powerful and cheap computational processing abilities, and affordable data storage has helped machine learning excel in this modern world. The applications of machine Learning ranges from email spam and malware filtering, fraud detection, to medical diagnosis, facial recognition, and autonomous cars. The acceptance of ML applications in our life has been swift. It has been an inseparable part of our daily life, and invisible at the same time.

The possibilities AI/ML bring in any industry space today is infinite. With the availability of vast amount of data being generated every second, AI systems are only getting better. However, the integrity of the data

should be preserved to get a fair result from ML systems. The data can be breached and tampered with during the training and after the training of the machine learning model. It is also of utmost importance to make sure the machine learning model itself is secure from attacks, as it can influence the model to make wrong decisions. Hence, it is absolutely necessary to take a pause and assess how secure and robust ML systems really are.

Security attacks have been prevalent in AI/ML as the application of AI has been increasing. Attackers are getting innovative and using sophisticated techniques to launch attacks. They are up-to-date with ever-changing technology space and are the first ones to take advantage of the fragile system. They can use disruptive technology to generate adaptive attacks, resulting in devastating results. AI systems can be utilized to launch attacks on a larger scale and target more victims. For instance, Deeplocker is a kind of AI-powered malware that can avoid detection by most of the security control in place today.

It has been proven time and again that it is possible to trick ML models with tainted data. As most ML algorithms are black-box algorithms, we do not know a lot about how they make decisions. As AI systems are prevalent and humans tend to easily trust the decisions made by these systems, it is important for us to protect AI systems from the attacks. An attacker produces adversarial samples as inputs to a machine learning model in order to cause the algorithm to make a mistake. Both the training and learning phases, as well as the operational phase, are good times to alter data presented to ML systems. Adversaries are eager to identify possible misclassified inputs or force the ML model to adopt undesirable behaviors, causing it to misclassify inputs on a regular basis. There are various sorts of assaults in adversarial machine learning. Data contamination attacks are essentially poisoning assaults. As a result, the model underperforms when it is deployed. Evasion is the most common sort of attack, in which the attacker manipulates data during deployment in order to fool a previously trained model. Model extraction is the process of an attacker exploring a black-box model in order to reproduce it or extract the data acquired from it.

On the pre-trained model, the attackers can essentially utilize two separate adversarial approaches (Fujimoto and Pedersen 2021). Untargeted adversarial attacks causes the model to identify incoming images inaccurately. As a result, the attack's objective is accomplished as long as the classification is incorrect. The attackers have no influence over the altered image's final output label. Targeted adversarial attacks, on the other hand, are designed in such a way that the attackers may choose and control the final anticipated label of the altered image. Not only does the model misclassify the image, but it also misclassifies the perturbed image with whatever the attacker chooses.

Adversarial instances just have few effective countermeasures. So far, two tactics have shown to be successful. Adversarial training is a brute-force approach in which a large number of adversarial cases are produced and fed into the system, with the model being trained to ignore them. Defensive distillation is a technique for training a model to provide probabilities for different classes. Before the training, the data cleaning approach is also used to discover and filter fraudulent data. The disadvantage is that determining what makes up malicious inputs might be challenging.

As hostile instances can be created in a number of ways, attacks against them are difficult to defend against. Because we can't put pen to paper to explain these scenarios, it's difficult to come up with a defense mechanism. It is also difficult to claim that these approaches will prevent a set of adversarial instances. Machine learning models generally generate impressive results, making adversarial attacks harder to counter. The circumstance in which the Adversarial example input effects the machine learning model is difficult to determine. In addition, the defense mechanisms aren't receptive to change. It may protect against one type of attack, but it also leaves the door open for attackers to exploit any weaknesses that aren't addressed.

This research shows how a machine learning model may be readily fooled into making incorrect classifications using one-pixel, multi-pixel, and all-pixel adversarial instances. The rest of the paper is laid out as follows. Section II describes the related work on establishing hostile instances in order to gain a better

understanding of their capabilities. The approach, dataset, and experiment setting are all covered in Section III. The experiment's results and findings are presented in Section IV. Section V is devoted to future work. Finally, in Section VI, the conclusions are presented.

2 RELATED WORK

An adversarial example is a data input containing a minor, deliberate modification that leads the machine learning model to predict incorrectly. The widespread use of AI systems has provided attackers with a lot of opportunities to confuse the systems and cause it to produce poor results. If a dirt splashes over a stop sign, for example, a self-driving automobile may be involved in an accident. The sign appears to be a stop sign to human eyes, but sign recognition software may categorize it as something different, causing the system to ignore the sign and crash the car. There are variety of ways to create adversarial examples. Some methods need access to the model's gradients, which is only possible with gradient-based models like neural networks, whilst others just necessitate access to the model's prediction function.

As this study focuses on adversarial instances on images, we'll go through some of the prior adversarial examples used to misclassify images. Adversarial images are images that have been purposefully manipulated in order to deceive the model during the application process. The following examples demonstrate how easily pictures that appear to be unaffected by human vision may trick deep neural networks for object detection.

(Szegedy et al. 2014) developed a gradient-based optimization strategy to find adversarial instances in deep neural networks. This approach transforms a picture into an array of pixels and then modifies the pixels to create an adversarial image. The difference between an adversarial image's predicted outcome and the desired incorrect outcome, as well as the difference between the original and adversarial instances, is calculated using the loss function. This can help bridge the gap between the adversarial and original images, allowing the prediction to default to the attack's intended label.

$$adv\_x = x + \varepsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

(Goodfellow, Shlens, and Szegedy 2015) created a method for generating adversarial images that involves adding or subtracting a minor inaccuracy to each pixel (1). While (Goodfellow, Shlens, and Szegedy 2015) required many pixels to be modified, (Su, Vargas, and Sakurai 2019) proved that we can generate an adversarial example by modifying just one pixel in a picture. Differential evolution is used in the one-pixel attack to select which pixel should be altered and how it should be changed (2).

$$x_i(g+1) = x_{r1}(g) + F(x_{r2}(g) - x_{r3}(g)), r1 \neq r2 \neq r3 \tag{2}$$

(Brown et al. 2017) devised a technique for deceiving AI systems by printing a label that can be put next to items to make them appear to be toasters for an image classifier. The adversarial image is created by replacing a piece of the original image with a patch of any shape. To make this technique adaptable, this patch may be rotated, relocated at various locations within the image, made larger or smaller, and replaced with a section of a different image.

(Athalye et al. 2018) built a 3D version of (Brown et al. 2017). To train a deep neural network, the authors 3D printed a turtle that looked like a gun from every angle. The authors devised a method for creating a 3D adversarial example for a 2D classifier that is adversarial across transformations such as all conceivable rotations, zooming in, and so on.

Without access to training data or internal model knowledge, (Papernot et al. 2016) proved that hostile instances can be constructed. Most adversarial attacks require access to the gradient of the underlying deep neural network to locate adversarial instances. (Papernot et al. 2016) , on the other hand, invented a black box attack, which is a zero-knowledge attack.

The research work presented in this paper is closely associated with the work presented in (Goodfellow, Shlens, and Szegedy 2015) and (Su, Vargas, and Sakurai 2019). As (Su, Vargas, and Sakurai 2019) exhibits the use of one-pixel to generate adversarial images, and (Goodfellow, Shlens, and Szegedy 2015) uses multiple pixels to generate them, this study analyzes and contrasts how these approaches vary and which method is more successful. The tests utilizing one-pixel and multi-pixel manipulated photos reveal how it affects model accuracy and effectively fools the machine learning system.

With all of the research being done to replicate the development of adversarial instances in order to better understand and protect our AI systems, creating an adaptable mechanism to generate security measures against such attacks is a tricky problem. These investigations provide researchers a better knowledge of how AI systems are vulnerable to adversarial attacks, which aids in the development of necessary security mechanisms.

3 RESEARCH APPROACH

3.1 Dataset

The dataset used for this experiment is ImageNet, from which a Chihuahua image is used to create Adversarial example to train on the Convolutional Neural Network(CNN), MobileNet. ImageNet is an image database arranged according to the WordNet hierarchy, with hundreds of thousands of photos depicting each node of the network (Deng et al. 2009). Mobilenet is a form of convolutional neural network built for mobile and embedded vision applications (Howard et al. 2017). The model can be manually trained using other images or the images from ImageNet. A pre-trained version of MobileNet from Keras, which has been trained on over a million photos from the ImageNet database, can also be utilized. For the purpose of this research, a pre-trained version of the model is being used.

3.2 Approach

Traditional image recognition techniques used color histograms and edge detection to categorize images made up of raw pixels. However, as the model was employed to predict photos with more complex attributes, the model began to fail. As a result, CNN was created to extract a higher level of representation for image content. The CNN model can train on the image's pixel and extract the features automatically for improved categorization.

The convolution operation picks a window in an image to analyze a subset of it. The pixel values for the window input and filter dot product are computed, which aids in focusing on important characteristics in an image. Convolution is used to create feature maps, which emphasize on the most essential features.

MobileNet model uses the alpha value of 1.0 which represents the default number of filters used at each layer in MobileNet (Howard et al. 2017). The weights of the model is based on ImageNet database. The dropout rate defaults at 0.0001. The number of classes to be classified into is set as 1000. The activation function used at the top layer is softmax.

The experiment involves generating perturbed Chihuahua image and using pre-trained CNN model, MobileNet to make predictions on the image. Transfer learning is a technique where existing model is used to

classify quickly. MobileNet is one of the popular example of Transfer learning. Keras library is being used to import MobileNet model. The image is preprocessed before feeding it to the model. The url for the image is sent as an argument to download from the cloud and write to a filename.

Millions of images in ImageNet are put together and assigned a label. Each label or node has thousands of pictures specific to that label. For example, Chihuahua is a label assigned to a number n02085620 and all the Chihuahua images belong in this class. All the url to the images belonging to this label can be generated by quering with the label number assigned to the label.

Once the image is read, it is preprocessed by resizing to (224, 224) which is the size of the images used to train the MobileNet model. The image is converted to an array and the dimension of the image is expanded to match the dimension of the images used to train the MobileNet model. The original image is used to make a prediction, to set a benchmark for our results analysis.

Figure 1: ImageNet Chihuahua image used as an adversarial example in the experiment.

$$P(x,y) \longrightarrow (x,y) : [r,g,b] + \Delta[r,g,b] \tag{3}$$

The perturb function P simply takes the (*x, y*) pixel and image as an argument and sets the selected (*x, y*) pixel to be the desired rgb value. rgb value is represented as [*r, g, b*] in an array. The rgb value used to perturb the image for this experiment is [255, 255, 0]. 0, signifies the absence of the specific color. Here, blue is absent from the rgb value array, which creates a yellow-colored pixel as a combination of red and green color.

The experiment is broken down into three parts using three different approaches. The first approach uses a one-pixel attack to perturb the image by just one pixel. The second approach involves multiple random pixels to be perturbed from the image, which covers 25% of the pixels in the image. Finally, the last approach involves perturbing all the pixels in the image. The experiment attempts to show how the prediction capability of the neural network gets affected by perturbing images with single, quarter, and all pixels in the image.

Figure 2 represents the flow diagram that summarizes the experiment. The image read is used to create versions of perturbed images, which are used to make predictions using MobileNet model.

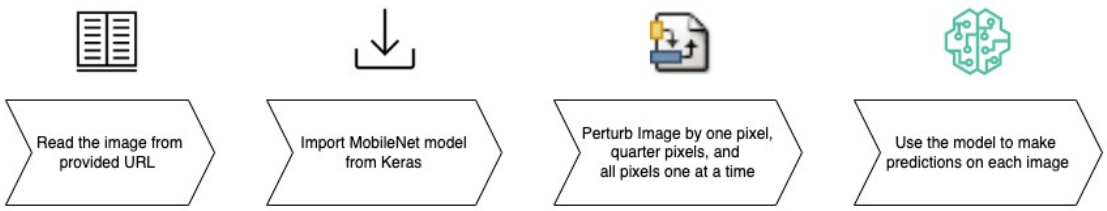

Figure 2: Flow diagram summarizing the experiment.

3.3 One-Pixel Attack

One-pixel attack (Su, Vargas, and Sakurai 2019) is a type of attack where a single pixel of the image is altered. A single-pixel modification doesn't make a lot of difference in the image visually, although it changes the prediction made by the MobileNet model.

Figure 3: ImageNet Chihuahua image with one-pixel perturbed.

A single pixel in the image is chosen at random to alter its rgb value. The rgb value of chosen (*x, y*) pixel is altered to have the [255, 255, 0] rgb value representing the color yellow. Although, this image is a part of the ImageNet dataset used to train MobileNet, due to the change in the rgb value of its pixel, the algorithm does show the change in the model accuracy.

3.4 Multi-Pixel Attack

Multi-pixel attack in the experiment is carried out by perturbing more than one pixel in the image. 25% of the pixels in the image are perturbed. The pixels chosen in the image can be a block of a continuous pixel in the image. However, perturbing continuous quarter pixels in an image can have different effects in different images. In some images, an insignificant part of the image where the primary subject does not lie might be perturbed, while in some images it might block the primary subject out. Therefore, we have chosen 25% of random pixels from the image to perturb, which results in a randomly generated adversarial image. Same as one-pixel attack, chosen (*x,y*) pixel values will have a $\Delta[r, g, b]$ changes to its rgb value to produce altered

Figure 4: ImageNet Chihuahua image with multi-pixel perturbed.

image. Figure 4 shows the adversarial image mostly unchanged to the human eyes except for some pixelated areas.

3.5 All-Pixel Attack

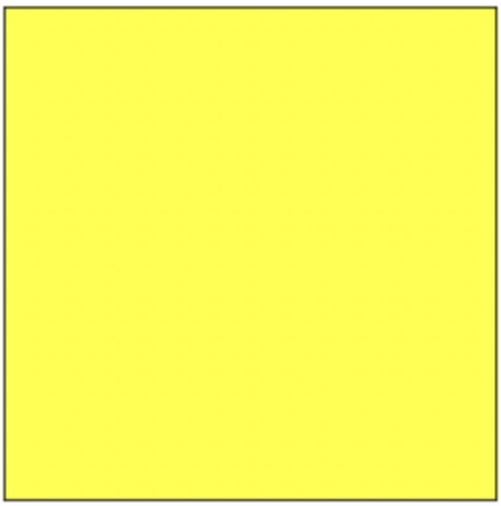

Figure 5: ImageNet Chihuahua image with all-pixels perturbed.

All-pixel attack involves perturbing the entire image with the rgb value of [255, 255, 0]. This will completely block the image and generate an image unidentifiable to human eyes as well. This image is only generated to compare with other perturbed images to show how significant impact it has on the predictions. All the pixel values (x,y) is altered with the $\Delta[r,g,b]$ which makes the image a complete yellow colored perturbed image.

4 FINDINGS

The predictions of the MobileNet model on the original image and the one-pixel perturbed image are shown in Figure 6 and Figure 7. The graphics depict the model's top five predictions, along with the degree of confidence it has in each one. From the original image to a one-pixel perturbed image, the model's confidence that the image is "Chihuahua" reduced from 0.917 to 0.913. The model's confidence in inaccurate predictions has also grown. "Carton," for example, was the second prediction with a confidence rating of 0.041 at first, but the one-pixel disturbed picture raised the carton prediction to 0.044. Similarly, with an undisturbed image, "miniature pinscher" was predicted with 0.017 confidence, which remained the same at 0.017 with the one-pixel perturbed image.

```
[[('n02085620', 'Chihuahua', 0.91693276),
  ('n02971356', 'carton', 0.041466217),
  ('n02107312', 'miniature_pinscher', 0.016510503),
  ('n02108915', 'French_bulldog', 0.005804981),
  ('n02096585', 'Boston_bull', 0.0054111257)]]
```

Figure 6: Predictions on unaltered Chihuahua image.

```
[[('n02085620', 'Chihuahua', 0.91371137),
  ('n02971356', 'carton', 0.043758582),
  ('n02107312', 'miniature_pinscher', 0.01696126),
  ('n02108915', 'French_bulldog', 0.0060314927),
  ('n02096585', 'Boston_bull', 0.005356382)]]
```

Figure 7: Predictions on one-pixel perturbed image.

Figure 8 and Figure 9 depict the change in prediction confidence in the perturbed picture with multi-pixels and all-pixels. The results show that with the multi-pixel perturbed image, the prediction confidence that there are Chihuahuas in the picture has reduced much further. With 25% of the pixels in the image changed, the confidence of "Chihuahua" has fallen to 0.875. The second-best prediction in the multi-pixel picture is "French_bulldog," which was fourth in both the original and one-pixel altered images.

```
[[('n02085620', 'Chihuahua', 0.87513745),
  ('n02108915', 'French_bulldog', 0.029640771),
  ('n02971356', 'carton', 0.028842859),
  ('n02107312', 'miniature_pinscher', 0.024846772),
  ('n02096585', 'Boston_bull', 0.0109809125)]]
```

Figure 8: Predictions on multi-pixels perturbed image.

```
[[('n03207941', 'dishwasher', 0.18689789),
  ('n02840245', 'binder', 0.08528389),
  ('n03982430', 'pool_table', 0.0629296),
  ('n04372370', 'switch', 0.0524962),
  ('n04548280', 'wall_clock', 0.03680243)]]
```

Figure 9: Predictions on all-pixels perturbed image.

Predictions based on the all-pixel altered image yield no useful information. The model is unable to recognize any objects in the picture since rgb [255, 255, 0] has totally occluded the image. The model's best guess based on the image is "dishwasher." "binder," "pool_table," "switch," and "wall_clock" are the next words

Table 1: Prediction on Chihuahua ImageNet Image for original and perturbed images.

Image Object	Original	One-pixel	Multi-pixel
Chihuahua	0.917	0.913	0.875
carton	0.041	0.044	0.029
miniature_pinscher	0.017	0.017	0.025
French_bulldog	0.006	0.006	0.030
Boston_bull	0.005	0.005	0.011

in the forecast. The all-pixel perturbed image's predictions aren't even comparable to the other modified pictures in the experiment.

The results reveal that the perturbed image categorization is definitely more incorrect as presented in Table 1. Even if the image seems unchanged to the human eye, the model began to demonstrate a decrease in the likelihood of prediction on the image. The model might entirely miss the prediction if the correct set of pixels are changed. The multi-pixel attack widened the gap and lowered the confidence in correct prediction, demonstrating that the more pixels are changed, the less exact the model becomes. The all-pixel assault was intended to demonstrate how incorrect the model is when compared to images with fewer pixels changed.

5 FUTURE WORK

The paper provides a straightforward explanation of how adversarial instances may be constructed and deployed against a system to push it to make wrong conclusions. To generate an adversarial example, a one-pixel untargeted assault is used as the attack type. Differential evolution can be used to construct one-pixel assaults. Differential evolution decides which pixels should be modified and how they should be changed. This will aid in the creation of a strong adversarial example.

This study may be expanded to look at how adversarial cases can be recognized and mitigated to help with machine learning security. The answer to minimize adversarial attacks is a more difficult issue, but it is critical for expanding the research horizon in this sector. It's difficult to come up with a strategy since the attacks are difficult to spot. Because the assaults are multi-variant, the need for an adaptive solution that works in the majority of adversarial scenarios can be considered as the top priority in the adversarial case's security aspect.

6 CONCLUSION

Adversarial instances demonstrate that the machine learning algorithms in which we place so much confidence and trust can be thwarted in spectacular ways. A little change in a pixel in a picture might lead the model to predict incorrectly. Adversarial strikes carried out in the incorrect location might have disastrous repercussions. Autonomous driving automobiles and medical care are the most susceptible areas, with potentially catastrophic outcomes.

These failures demonstrate that even a minor change in the environment can lead the algorithm to react in a way that the creators did not intend. The majority of machine learning (ML) solutions on the market today are black box appliances that are installed in networks to ingest data, process it, and make choices without requiring people to know what the model is doing. Data taint will always be a concern throughout the training or operating phases, and it will always constitute a hazard. It's nearly hard to be certain of the origin and integrity of enormous amounts of data. As a result, enemies have a plethora of vectors at their disposal to modify data.

The experiments in this work were conducted to construct adversarial cases in order to deceive the AI system. It demonstrated how ML systems may be readily fooled by changing one or more pixels in a picture while people are unable to detect the changes. With increasing pixel augmentation, the model's prediction skill deteriorates. Effectively working toward discovering solutions to adversarial cases is of paramount relevance and necessity. To bridge the gap between what designers want and how algorithms act, it's vital to be involved and assist establish solutions for preventing adversarial situations.

ACKNOWLEDGMENTS

This research was funded by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory, and in part by the US NSF grant CNS/SaTC 2039583 and Mastercard Research Funds at Howard University. However, any opinion, findings, conclusions, or recommendations expressed in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

REFERENCES

Alhajjar, E., P. Maxwell, and N. D. Bastian. 2021. "Adversarial Machine Learning in Network Intrusion Detection Systems". *Expert Syst. Appl.* vol. 186, pp. 115782.

Athalye, A., L. Engstrom, A. Ilyas, and K. Kwok. 2018, 10–15 Jul. "Synthesizing Robust Adversarial Examples". In *Proceedings of the 35th International Conference on Machine Learning*, Volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293, PMLR.

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey, Prentice-Hall, Inc.

Brown, T. B., D. Mané, A. Roy, M. Abadi, and J. Gilmer. 2017. "Adversarial Patch". *ArXiv* vol. abs/1712.09665. Available via http://arxiv.org/abs/1712.09665. Accessed Jun. 30, 2022.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A large-scale hierarchical image database". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.

Fujimoto, T., and A. P. Pedersen. 2021. "Adversarial Attacks in Cooperative AI". *ArXiv* vol. abs/2111.14833. Available via https://arxiv.org/abs/2111.14833. Accessed Jun. 30, 2022.

Goodfellow, I. J., J. Shlens, and C. Szegedy. 2015. "Explaining and Harnessing Adversarial Examples". In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun.

Gu, S., and L. Rigazio. 2015. "Towards Deep Neural Network Architectures Robust to Adversarial Examples". In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, edited by Y. Bengio and Y. LeCun.

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. 2017. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". *ArXiv* vol. abs/1704.04861. Available via http://arxiv.org/abs/1704.04861. Accessed Jun. 30, 2022.

Lin, J., L. Dang, M. Rahouti, and K. Xiong. 2021. "ML Attack Models: Adversarial Attacks and Data Poisoning Attacks". *ArXiv* vol. abs/2112.02797. Available via https://arxiv.org/abs/2112.02797. Accessed Jun. 30, 2022.

Merrigan, A., and A. F. Smeaton. 2021. "Using a GAN to Generate Adversarial Examples to Facial Image Recognition". *ArXiv* vol. abs/2111.15213. Available via https://arxiv.org/abs/2111.15213. Accessed Jun. 30, 2022.

Papernot, N., P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. 2016. "Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples". *ArXiv* vol. abs/1602.02697. Available via http://arxiv.org/abs/1602.02697. Accessed Jun. 30, 2022.

Su, J., D. V. Vargas, and K. Sakurai. 2019. "One Pixel Attack for Fooling Deep Neural Networks". *IEEE Transactions on Evolutionary Computation* vol. 23 (5), pp. 828–841.

Sun, H., T. Zhu, Z. Zhang, D. Jin, P. Xiong, and W. Zhou. 2021. "Adversarial Attacks Against Deep Generative Models on Data: A Survey". *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. 2014. "Intriguing properties of neural networks". *ArXiv* vol. abs/1312.6199. Available via http://arxiv.org/abs/1312.6199. Accessed Jun. 30, 2022.

Vargas, D. V., and J. Su. 2019. "Understanding the One-Pixel Attack: Propagation Maps and Locality Analysis". *ArXiv* vol. abs/1902.02947. Available via http://arxiv.org/abs/1902.02947. Accessed Jun. 30, 2022.

Zhang, J., W. J. Tann, and E. Chang. 2021. "Mitigating Adversarial Attacks by Distributing Different Copies to Different Users". *ArXiv* vol. abs/2111.15160. Available via http://arxiv.org/abs/2111.15160. Accessed Jun. 30, 2022.

AUTHOR BIOGRAPHIES

UTSAB KHAKUREL is a Ph.D. Student in the Department of Electrical Engineering and Computer Science (EECS) at Howard University, Washington, DC, USA. His research interest lies in bias ML, explainable AI, cybersecurity, computer vision, Internet-of-Things and other related topics in AI/ML and cybersecurity. His research focuses to find solutions on AI for Cybersecurity and Cybersecurity for AI. His email address is utsab.khakurel@bison.howard.edu.

DANDA B. RAWAT is the Associate Dean for Research Grad Education in College of Engineering, a Full Professor in the Department of Electrical Engineering Computer Science (EECS), Founding Director of the Howard University Data Science Cybersecurity Center, Founding Director of DoD Center of Excellence in Artificial Intelligence Machine Learning (CoE-AIML), Director of Cyber-security and Wireless Networking Innovations (CWiNs) Research Lab, Graduate Program Director of Howard CS Graduate Programs and Director of Graduate Cybersecurity Certificate Program at Howard University, Washington, DC, USA. Dr. Rawat is engaged in research and teaching in the areas of cybersecurity, machine learning, big data analytics and wireless networking for emerging networked systems including cyber-physical systems (eHealth, energy, transportation), Internet-of-Things, multi domain operations, smart cities, software defined systems and vehicular networks. He has secured over $20 million in research funding from the US National Science Foundation (NSF), US Department of Homeland Security (DHS), US National Security Agency (NSA), US Department of Energy, National Nuclear Security Administration (NNSA), DoD and DoD Research Labs, Industry (Microsoft, Intel, PayPal, Mastercard, Meta, BAE, Raytheon etc.) and private Foundations. Dr. Rawat is the recipient of NSF CAREER Award in 2016, Department of Homeland Security (DHS) Scientific Leadership Award in 2017, Provost's Distinguished Service Award 2021, Researcher Exemplar Award 2019 and Graduate Faculty Exemplar Award 2019 from Howard University, the US Air Force Research Laboratory (AFRL) Summer Faculty Visiting Fellowship 2017, Outstanding Research Faculty Award (Award for Excellence in Scholarly Activity) at GSU in 2015, the Best Paper Awards (IEEE CCNC, IEEE ICII, IEEE DroneCom and BWCA) and Outstanding PhD Researcher Award in 2009. He has delivered over 50 Keynotes and invited speeches at international conferences and workshops. Dr. Rawat has published over 200 scientific/technical articles and 11 books. He has been serving as an Editor/Guest Editor for over 100 international journals including the Associate Editor of IEEE Transactions of Service Computing, Editor of IEEE Internet of Things Journal, Associate Editor of IEEE Transactions of Network

Science and Engineering and Technical Editors of IEEE Network. He has been in Organizing Committees for several IEEE flagship conferences such as IEEE INFOCOM, IEEE CNS, IEEE ICC, IEEE GLOBECOM and so on. He served as a technical program committee (TPC) member for several international conferences. He served as a Vice Chair of the Executive Committee of the IEEE Savannah Section from 2013 to 2017. Dr. Rawat received the Ph.D. degree from Old Dominion University, Norfolk, Virginia. Dr. Rawat is a Senior Member of IEEE and ACM, a member of ASEE and AAAS, and a Fellow of the Institution of Engineering and Technology (IET). He is an ACM Distinguished Speaker (2021- 2023). His email address is danda.rawat@howard.com.